IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Cost-based heterogeneous learning framework for real-time spam detection in social networks with expert decisions

Jaeun Choi[1] and Chunmi Jeon[2]

[1]Department of Artificial Intelligence Software, Kyungil University, 50, Gamasil-gil, Hayang-eup, Gyeongsan-si, Gyeongsangbuk-do 38428, Republic of Korea

[2]School of Business and Technology Management, College of Business, Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea

Corresponding author: Jaeun Choi (e-mail: juchoi@kiu.kr).

ABSTRACT With the widespread use of social networks, spam messages against them have become a major issue. Spam detection methods can be broadly divided into expert-based and machine learning-based detection methods. When experts participate in spam detection, the detection accuracy is fairly high. However, this method is highly time-consuming and expensive. Conversely, methods using machine learning have the advantage of automation, but their accuracy is relatively low. This paper proposes a spam-detection framework that combines and fully exploits the advantages of both methods. To reduce the workload of the experts, all messages are first analyzed via a primary machine learning filter, and those that are determined to be normal messages are allowed through, whereas suspicious messages are flagged. The flagged messages are subsequently analyzed by an expert to enhance the overall system accuracy. In the filtering process, cost-based machine learning is used to prevent the fatal error of misidentifying a spam message as a normal message. In addition, to obviate the continuously evolving spam trends, a module that periodically updates the expert-diagnosis results on the training dataset is incorporated into the framework. The results of experiments conducted, on an imbalanced dataset of spam tweets and normal tweets in a ratio similar to the actual situation in real life, indicate that the proposed framework has a spam-detection rate of almost 92.8%, which is higher than that of the conventional machine learning technique. Furthermore, the proposed framework delivered stable high performance even in an environment where social network messages changed continuously, unlike the conventional technique, which exhibited large performance deviations.

INDEX TERMS Expert decision making, machine learning, real-time spam detection, social network, Twitter spam.

I. INTRODUCTION

The number of Internet users globally is estimated to be approximately 4.9 billion, which is approximately 63% of the global population of 7.7 billion [1]. Many social network companies compete fiercely in this market. Social networks that allow users to communicate anytime and anywhere are becoming a part of everyday life for many people around the world, particularly among the 3.5 billion smartphone users worldwide [2]. According to Visual Capitalist, a market research firm in the US, the number of monthly active users of Facebook, the social network that had the largest number of users in 2020, is as many as 2.6 billion. Further, Instagram and Twitter have monthly active users of approximately 1 billion and 0.3 billion, respectively [3]. As social networks have become a part of everyday life for many people, attacks targeting them are also posing serious threats. In particular, spam messages in social networks instill political prejudices, disrupt the stock market, steal personal information, or spread false information [4]. In fact, the spreading of fake news is accelerated by autobots acting on Twitter, which results in such news spreading faster than normal news on the network [5]. Moreover, the spreading of advertisements for illegal products often occurs [6]. It has been found that one out of every 21 tweets on Twitter can be categorized as spam, and autobots account for approximately 15% of Twitter users [7]. The spam distributed through Twitter are known to be more dangerous than general spam. The click-through rate of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098799, IEEE Access

general spam mails is only 0.0003–0.0006%, whereas the click-through rate of Twitter spam is as high as 0.13% [8].

Recently, many accounts have spread false information and spam related to the 2020 US presidential election [9]. In July 2020, more than 130 famous accounts, including those of Barack Obama, Joe Biden, and Elon Musk, were stolen, and fake Bitcoin transactions were tweeted from them [10]. Thus, spam spreading through social networks such as Twitter continuously pose threats. This paper proposes a method for detecting such spam.

A variety of methods to detect spam on Twitter have been proposed and are being used. The most reliable method is to detect spam through the participation of experts. The blacklist technique, one of the most representative detection methods, involves the addition of malicious URLs to a blacklist. Tweets that contain the blacklisted URLs are then blocked. However, one of the disadvantages of this technique is its inability to block new spam because it takes some time to update the blacklist [8]. Experts can install honeypots at suitable locations on social networks and detect spam accounts [11, 12]. Honeypots are systems installed intentionally to induce attackers and are widely used to detect attacks and spam in networks. Malicious accounts on social networks can also be found through crowdsourcing by experts and general participants together [13]. When experts participate in detection, the accuracy can increase significantly. However, direct human participation has the disadvantage of being time-consuming and costly.

To detect spam automatically without assistance from experts, many methods using machine learning have been proposed. Machine learning is widely used in various fields for automatic learning and detection as well as for detecting spam on Twitter. Previous studies have extracted and used various features ranging from simple to complex and also used various learning and classification algorithms ranging from traditional machine learning techniques to deep learning. Although there are some differences in the features and algorithms used, many studies have found that the performance of spam detection using machine learning is not poor. However, the detection accuracy falls even if the attacks are slightly transformed because machine learning relies solely on existing data. In fact, an analysis of spam revealed that the characteristics of spam change continuously [14]. In other words, it is difficult to respond to new types of attacks using machine learning that learns only from existing data. Furthermore, the results can vary greatly depending on the selected feature set. As there are many types of spam tweets, it is often impossible to detect them by relying on specific features. Hence, if an incorrect feature set is selected, or the attacker ascertains which feature set is used in the training and avoids it, the accuracy of detection decreases. Furthermore, whereas most studies perform training with normal and spam data in a 1:1 ratio, because the ratio of spam is much smaller in reality, training should be performed with a dataset that reflects this fact. A few studies have attempted to solve this imbalanced data problem using a data sampling technique. However, the results of this technique may not reflect reality because in this technique a small amount of the collected data is used to represent an entire class. Thus, systems for automatically detecting spam without human assistance paradoxically have the problem of not properly filtering out spam when the machine is operated fully autonomously.

This paper proposes a heterogeneous framework in which experts and machines detect spam in conjunction. The biggest problem in using expert-based decisions is that it is time-consuming. This problem can be alleviated by preventing an overload on the experts. To prevent such an overload, we first filter out normal tweet data that can be clearly distinguished through machine learning. By adjusting the cost of the classification errors, the definite normal data are filtered out first, and only the suspicious data are sent to the experts for inspection. Consequently, a high detection rate can be achieved while preventing overload on experts. Furthermore, the diagnosis results of experts can be periodically applied to update and train the primary filter to respond to evolving attacks.

The remainder of this paper is organized as follows. Section II gives an overview of existing studies on spam-detection techniques using human-assisted approaches and machine learning techniques. Section III examines in detail the problems encountered when using only the machine learning technique and presents the justification for incorporating expert participation in spam detection. Section IV introduces the proposed Twitter spam-detection method that combines machine learning with the expert decisions. Section V outlines the experiments conducted and analyzes the experimental results obtained. Finally, Section VI presents concluding remarks and outlines future research directions.

II. RELATED WORK

A. HUMAN-ASSISTED APPROACH

Many spammers distribute messages that contain URLs for malicious purposes. Therefore, the method of adding malicious URL links in the blacklist and blocking messages that contain the blacklisted URLs is widely used. Twitter blocks malicious links using Google Safe Browsing API [8]. However, one disadvantage of this method is its inability to block new spam because it takes about four days to add new spam URLs to the blacklist [15]. Furthermore, detecting spam by using only the blacklist method is inadequate because many spammers spread spam using shortened URLs [16].

One method used by experts to detect malicious actions directly is the use of honeypot. Widely used to respond to conventional attacks on networks, the honeypot creates a virtual server to lure and detect intruders who attack it. This approach is used extensively to detect spam mails. Methods using honeypots to detect spam on social networks have been proposed as well. Lee et al. [11] detected and analyzed 36,000 spam tweet accounts using 60 honeypots over a seven-month

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098799, IEEE Access

period. Stringhini et al. [12] installed honey-profiles for three social networks (Facebook, Twitter, and MySpace) to detect and analyze spam.

The "Social Turing Test" method has also been suggested to detect malicious users on social networks using the crowdsourcing model, in which experts and general users participate [13]. According to this model, experts and participants, called turkers, detect malicious accounts on Facebook and Renren. Turkers with low accuracy are removed from the system, and the results from the turkers with higher accuracy are reflected more.

Thus, various spam-detection methods in which some or many experts participate have been proposed. These methods have exceptionally high accuracy because experts directly participate and make decisions. They are used in real environments because of their advantages. However, it takes much time to gather the opinions of experts, and some parts are operated manually. Therefore, techniques for automatically detecting spam using machine learning are being actively researched.

B. MACHINE LEARNING APPROACH

Many studies have been conducted using machine learning to detect spam automatically. As there are various types of information that can be extracted from social networks, such as Twitter, many studies have utilized various feature sets. The most commonly used features include information that can be extracted from Twitter accounts and tweet content. Furthermore, some researchers have analyzed the text information of the spread messages or the social graph that represents the relationships between Twitter accounts.

Many techniques for detecting spam after extracting statistical information that can be obtained from the tweets as features have been proposed by researchers. Some spammers spread spam using a technique that transforms malicious content into tweets [16]. Hence, spam can be detected using the characteristics that appear in the spam tweets as features. For example, spam tweets have more hashtags or URLs than normal tweets and contain spam words. Thus, spam can be detected by using features such as hashtags, URLs, and spam words [17]. In addition, techniques using the characteristic that spammers insert more numbers in spam tweets than in normal tweets have been used [18]. Chen et al. analyzed 6 million tweets and presented the number of tweets posted by users, number of retweets, number of hashtags, number of user mentions, number of URLs included in the tweets, weight of URLs, number of characters, and number of digits included in tweets as features [19]. The features that can be obtained from the tweets themselves are used in various studies because they intuitively reflect the characteristics of spam attacks and can be extracted relatively easily.

Although spam actions are performed by tweets, spam tweets are spread by spammers. Thus, statistical information on accounts related to spammers is also widely used as a feature. Spammers are generally characterized by opening their account relatively recently and having a small number of users following them compared with normal users. Account information that is mainly used as features includes the year of opening the account, number of followers, and reputation of the account [16]. The statistics of the accounts together with the statistics of the tweets mentioned above are easy to extract, and they considerably reflect the characteristics of spam. Thus, many studies have proposed using both feature sets in conjunction [19–21].

In addition to the account information of users, the text of tweets is also being actively analyzed and used as features. Studies have extracted the characteristics of spam tweets using the term frequency-inverse document frequency (TF-IDF) [22] and bag-of-words, which are widely used in text analysis [23–25]. Inuwa-Dutse et al. [26] analyzed keywords widely used in spam using latent semantic analysis (LSA) and used them as features. A study has also analyzed tweet content and metadata information with long short-term memory (LSTM) and deep neural networks [27].

Other notable studies have used the abovementioned features in combination with new features. A method has been proposed to detect spam accounts and spam tweets, and it involves extracting account information, tweet information, and n-gram of tweets as features, and then using multiple convolutional neural networks (CNNs) in an ensemble [28]. Sedhai and Sun [29] used hashtags and tweet content as features, considering that spam tweets contain many hashtags. They also proposed a semi-supervised methodology that additionally used domain addresses as features [30]. One study extracted features considering the year of creation of the account and the community in which the account was active and responded to spam tweets in real time by applying an unsupervised learning framework [15]. In addition, social graph-based techniques that detect spam accounts by analyzing the relationships between the following and followers of accounts have been proposed [31–33].

Detection techniques using machine learning have the advantage of being automated, even though the features and techniques used are different. However, most studies depend on supervised learning, which is static and requires experts to annotate each set of data. It is ironic that machine learning techniques that seek to minimize human intervention incur high expert-labor costs to prepare datasets [15]. Techniques using machine learning have various problems in addition to the fundamental problem of having to be preceded by human effort. We examine these problems in detail in Section III.

III. PROBLEM STATEMENT

A. DEPENDENCE ON FEATURES

As described in Section II.B, the features used by the machine learning techniques for detecting Twitter spam are statistical data that can be obtained from the tweets themselves. These features reflect the characteristics of attacks, such as spam tweets containing many numbers or hashtags. However, recent

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098799, IEEE Access

spam attacks often do not have any tags [34]. Attackers continuously generate attacks to avoid spam-detection techniques because they know the features used in the techniques. If features are generated only by analyzing old spams, they are inevitably vulnerable to new attacks. Moreover, if the statistical information of spam tweets is similar to that of normal tweets, attacks cannot be detected only by statistical features. Sometimes, statistical information can lead to the misidentification of normal users as spam users. For example, attackers tend to send many identical tweets simultaneously, and features reflecting this property are sometimes used. However, normal users also often send identical tweets simultaneously. In this case, normal users can be mistaken for spam users [35].

As the use of statistical information of tweets exclusively has limitations, we examined studies that also used account information as features. Users who spread spam predominantly have a small number of followers or their accounts have just been created. Thus, it is efficient to use account information. However, as every rule has an exception, spam is sometimes spread through old accounts or accounts with many followers [35]. Hackers, including spammers, tend to attempt to commit malicious acts through an account with many followers. In fact, by using the term "Twitter account sell" for a search, we easily found transactions for accounts with a set price per follower and a message stating that an account with 5,000 followers is traded for $200 at four cents per follower [36]. Thus, if features that can be used to detect many types of spam are exposed to spammers, methods to avoid them would easily appear.

Detection techniques using social graphs also show vulnerability to spammers disguised as normal users [16]. Furthermore, it takes much effort to analyze the features of detection techniques that use social graphs or analyze the contents of tweets using TF-IDF or bag-of-words. Attackers can also avoid these techniques sufficiently. For example, if a spammer sends spam messages that are highly similar to normal messages, it is difficult to detect them only by analyzing the text of the messages.

Various feature sets have been proposed for spam detection, but they have their respective disadvantage. They cannot filter out all the spam if they depend on specific features to detect spam. When all features are used in combination, considerable effort is required to create features, which is not consistent with the purpose of automation through machine learning techniques. This creates a paradoxical situation in which great effort is required from experts to create features and data, although machine learning works automatically.

B. IMBALANCED DATA

A binary classification problem, in which a very small number of samples corresponds to one class and a large number of samples corresponds to another class, can be observed frequently in real environments [37]. The machine learning research community denotes this as "class imbalance" and regards it as a challenging problem [38]. Twitter spam is no exception. Approximately 3% of all tweet messages exhibit malicious abuse behaviors by spammers [21]. If training and discrimination are performed with a dataset consisting of a large amount of normal Twitter data and a small amount of spam Twitter data, the spam-detection performance will invariably deteriorate because of data imbalance [39].

However, many methods that detect spam on Twitter overlook this class imbalance problem [37]. Several methods perform training and detection using a dataset composed of normal and spam data in a 1:1 ratio, which is far from the real situation. Hence, these methods do not account for the actual environment in which spam comprise only a small proportion. When training is performed with a dataset composed of a small amount of spam data and a large amount of normal data, similar to the actual environment, the spam-detection rate falls sharply even though the detection rate for normal data, which comprise the majority, is high [16].

To verify the effect of the imbalance of Twitter spam data on the accuracy of detection, we conducted an analysis using the dataset created for research by Chen et al. [19]. Collected through Twitter API, this dataset has been used as a reference in many studies on Twitter spam. By using this dataset, our study also enhanced the transparency. We will discuss the details related to the datasets and evaluation metrics in Section V.A. There are two Twitter spam datasets: one dataset is composed of normal and spam data in a 1:1 ratio, and the other is composed of normal and spam data in a 95:5 ratio. In this section, we refer to the dataset with a 1:1 ratio as dataset 1 and the dataset with a 95:5 ratio as dataset 2.

To study the effect of data imbalance, we applied various machine learning and deep learning techniques, which are used as basic classifiers in many studies, and compared the results between dataset 1 and dataset 2. The machine learning techniques used here were tree-based J48, random forest (RF), rule-based PART, Naïve Bayes network, and multilayer perceptron (MLP), which is a simple type of deep learning technique.

As shown in Table I, there are some differences in accuracy depending on the machine learning technique, but in the case of dataset 1, the detection accuracy for normal and spam data is high. In contrast, in the case of dataset 2, which has an imbalance problem, as the normal data account for 95%, the normal data detection accuracy and total accuracy are higher than those for dataset 1. However, the detection accuracy for spam is significantly lower. In particular, the recall value indicating whether spam data were correctly distinguished is lower by 5% at the minimum and more than 30% at the maximum. The low detection accuracy for spam highlights the issue with existing machine learning-based algorithms. A high detection rate for spam must be achieved even when training with data of the same composition as dataset 2 in order to detect spam on real social networks.

TABLE I
COMPARISON OF THE PERFORMANCE OF VARIOUS MACHINE LEARNING ALGORITHMS FOR DATASETS 1 AND 2

Machine Learning Algorithms		Dataset 1 (Normal:Spam = 1:1)			Dataset 2 (Normal:Spam = 95:5)		
		Precision	Recall	F-Measure	Precision	Recall	F-Measure
J48	Spam	0.942	0.925	0.934	0.930	**0.866**	0.897
	Normal	0.927	0.943	0.935	0.993	0.997	0.995
	Weighted Avg.	0.935	0.934	0.934	0.990	0.990	0.990
Random Forest	Spam	0.976	0.937	0.956	0.975	**0.883**	0.926
	Normal	0.940	0.977	0.958	0.994	0.999	0.996
	Weighted Avg.	0.958	0.957	0.957	0.993	0.993	0.993
PART	Spam	0.938	0.921	0.989	0.923	**0.841**	0.880
	Normal	0.922	0.939	0.930	0.992	0.996	0.994
	Weighted Avg.	0.930	0.930	0.930	0.988	0.989	0.988
Naïve Bayes	Spam	0.921	0.834	0.875	0.603	**0.717**	0.644
	Normal	0.848	0.928	0.887	0.985	0.975	0.980
	Weighted Avg.	0.885	0.881	0.881	0.966	0.962	0.964
MLP	Spam	0.900	0.870	0.885	0.890	**0.563**	0.690
	Normal	0.875	0.903	0.889	0.977	0.996	0.987
	Weighted Avg.	0.887	0.887	0.887	0.973	0.975	0.972

To solve the class imbalance problem that appears in Twitter spam, some methods perform training and classification after artificially resampling the dataset. A typical example is the oversampling method, which solves data imbalance by artificially sampling the minority class to make it similar to the sample number of the majority class [37]. The most widely used oversampling method is synthetic minority class over sampling (SMOTE), which resamples a number of synthetic data samples based on the data of the minor class, making the number of samples in the minor class similar to that of the major class [40]. As the class imbalance can be solved by cleaning data in advance, such as SMOTE, some studies have applied these techniques to detect Twitter spam [26,27]. However, if resampling is performed based on a small amount of data, the minor class will only have the characteristics of the small amount of collected data. Furthermore, this technique has the critical drawback that new test data cannot be detected [41]. In other words, the training and detection through data resampling are not effective for social networks in which new types of spam attacks continuously appear.

Thus, detecting spam using only machine learning cannot be free of the class imbalance problem. The machine learning approach is not capable of detecting spam or responding to new types of spam properly. Furthermore, as described above, the detection performance varies depending on the features used. Our study proposes a framework that reflects expert opinions to compensate for the disadvantages of using machine learning alone for spam detection.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098799, IEEE Access

C. SPAM DRIFT

Most machine learning techniques create a dataset from the existing collected data, which are annotated by an expert, and use it for training. The detection accuracy for data on which the techniques have not been trained is low. Twitter spam is no exception, and attackers continuously create new types of attacks to respond to new detection techniques. The characteristics of Twitter spam change over time—a phenomenon called "spam drift" by Chen et al. [14, 42]. According to the study, even in a relatively short period of 10 days, the ages of accounts sending spam tweets ranged from 530 to 730 days. This indicates a large change in spam data, considering that the ages of normal accounts during the same period ranged from 710 to 740 days. Such spam drifts can also be observed in features such as the number of following and the number of mentions per user. Therefore, responding to spam using only the existing machine learning approach, which has a weakness in detecting new types of spams, can cause serious problems.

Chen et al., who first raised the problem of spam drift, suggested a method of reflecting the data collected online in the training process through an asymmetric learning approach [42]. They also suggested a method of training based on tweets that have not yet been labeled and used for classification [14]. To respond to spam that evolve in real time, a framework that combines unsupervised classification and supervised learning algorithms has been proposed as well [15].

Although some studies have proposed methods to detect spam that evolve in real time, most studies that use machine learning ignore this problem and detect spam after training based on existing collected data. To respond to new spam, our study proposes a framework with a module that incorporates the classification results of experts in the new training process at regular intervals.

IV. DESIGN OF PROPOSED FRAMEWORK

A. MODEL DESIGN: AN OVERVIEW

As explained above, it is impossible to detect spam tweets perfectly by using machine learning only, and it is also difficult to detect evolving spams. This paper proposes a framework that combines expert opinion and machine learning. As shown in Fig. 1, our framework is composed of three steps: (i) cost-based tweets filtering, (ii) spam tweets detection with expert decisions, and (iii) training data update. First, to reduce the workload of experts, normal data that can be clearly distinguished are first filtered out through a machine learning technique. To filter out only the normal data with high certainty, we use a cost-based machine learning classifier. After filtering out normal data, the expert tests the tweets in depth for suspicious data only. The tests performed by experts generally take a long time but have a very high accuracy. If the training data are updated with the result of the final classification by an expert at predefined time intervals, our framework can respond to newly generated attacks as well. The details of each step are presented in the following sections.

B. COST-BASED TWEETS FILTERING

The first step in the proposed framework is to distinguish between normal and spam tweets using machine learning. To do this, numerous tweets that are streamed on the Internet are collected, and the features are then extracted from these data.

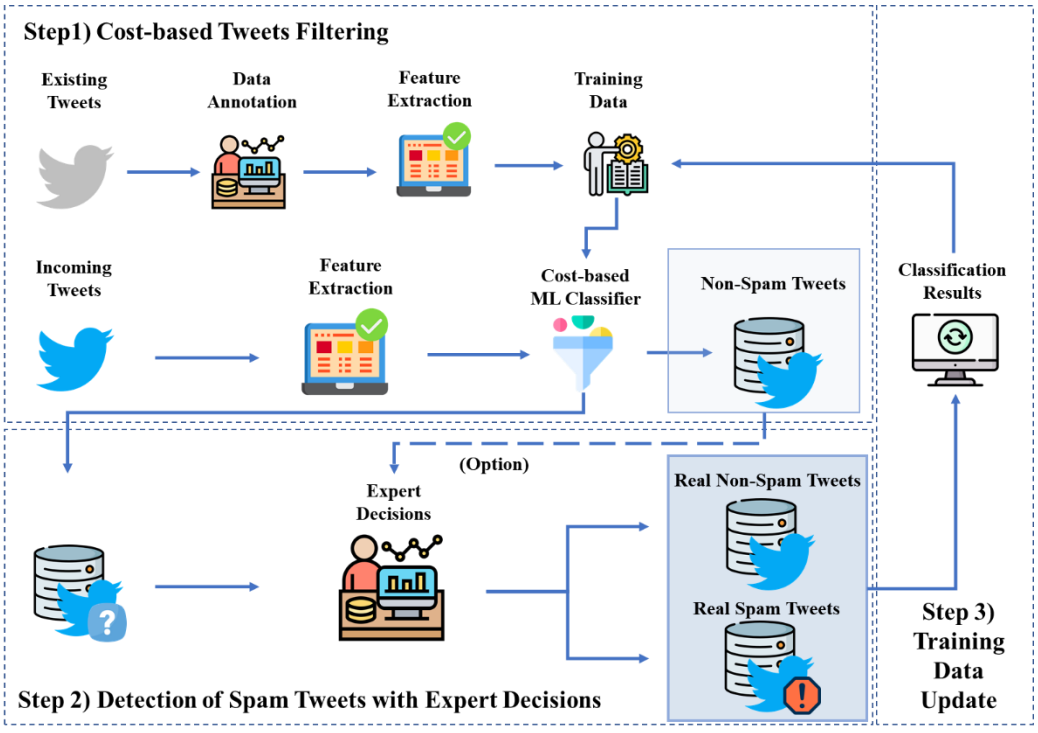

FIGURE 1. Workflow of the proposed framework

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098799, IEEE Access

By applying a machine learning algorithm to the extracted features, spam and non-spam tweets can be distinguished. In most spam-detection methods, the workflow ends when the spam are classified through the machine learning step. However, as mentioned above, perfect spam detection cannot be achieved using machine learning alone. Thus, the machine learning step in our proposed framework performs filtering rather than the final classification. In the filtering step, only the normal tweets are filtered through; the suspicious tweets are sent to the second step, in which they are inspected by an expert.

Unlike other techniques that end with the final machine learning classification, our proposed framework provides another opportunity for inspection in the next step. Therefore, although detection accuracy is important for the machine learning algorithm used in the filtering step, it is more important to prevent errors that misclassify spam tweets as normal. If normal tweets are misclassified as suspicious tweets, they can be corrected by the expert in the next step. However, if spam tweets are misclassified as normal by the first filter, they will end up as misclassified if the next step is not used. Therefore, although it is important for the filter to filter through as many normal tweets as possible, it is more important to prevent spam from being mixed with the normal tweets. To prevent misclassifications, which can be fatal to the filter, we use a cost-based machine learning technique. This method sets a different cost for errors that occur in the case of misclassification and attempts to minimize the sum of costs [43, 44]. The asymmetric classification cost matrix is presented in Table II.

TABLE II
ASYMMETRIC MISCLASSIFICATION COST MATRIX

	Actual Positive	**Actual Negative**
Predicted Positive	$C(p,p)$	$C(n,\ p)$
Predicted Negative	$C(p,n)$	$C(n,n)$

The cost-based classification uses an asymmetric misclassification cost matrix, as described above. In this study, spam, which comprise a minority in the dataset, are considered positive, and normal tweets are considered negative. There are two types of misclassification: (1) misclassifying positive spam as negative, and we denote the cost that occurs in this case $C(p,n)$; (2) misclassifying a negative normal tweet as positive, and we denote the cost that occurs in this case as $C(n,p)$. Cost-based detection focuses on minimizing the cost resulting from misclassification. Therefore, the case of judging a spam tweet as a normal tweet must be avoided. Thus, $C(p,n)$ is set higher than $C(n,p)$ to avoid the corresponding errors as much as possible. The cost $C(p,p)$ that corresponds to a true positive for detecting a spam as spam and the cost $C(n,n)$ that corresponds to a true negative for detecting normal tweets as normal must be set to zero because the cost is only generated by misclassification [43].

Once the cost settings are complete, the class to which a given sample belongs can be determined. To do this, the probability that a given sample belongs to a certain class is calculated first. The probability that an example x belongs to class j is as follows [43].

$$P(j|x) = \frac{1}{\sum_i 1}\sum_i P(j|x, M_i),$$

where i has a range of 1 to m, and m is the number of resamples newly generated by dividing all the samples. The number of examples for each resample is n. If S_i is a resample that has n examples, M_i is a model generated by applying the machine learning algorithm to S_i. Here, the risk that results when x belongs to class k can be defined as follows [44]:

$$R(k|x) = \sum_j P(j|x)C(k,j)$$

The risk generated when an example is allocated to a class must be minimized. Hence, class k that satisfies the following equation becomes the class to which x will be allocated:

$$argmin_k R(k|x)$$

In the case of spam detection, where there are two classes, class k can be set to spam or normal.

In the case of cost-based classification, the larger the cost of misclassification, the less the occurrence of the corresponding errors. Therefore, in this study, we set $C(p,n)$ to a high value so that spam would not be judged as normal. However, if the cost is set too high, many normal tweets will be also classified as suspicious tweets and will be subjected to expert diagnosis. In other words, if the cost is set high, fatal errors can be avoided, but the load on the expert increases. In contrast, if the cost is low, it becomes very similar to conventional machine learning classification. As a result, the number of tweets handed over to step 2 decreases, but the cases in which spam tweets are classified as normal tweets increases. Fig. 2 shows the filtering situation according to the cost setting. Even with the same cost-based classification algorithm, different results can be obtained depending on the cost setting. Thus, it is crucial to set appropriate costs in accordance with the situation.

More effective filtering is possible by adjusting the cost according to the given situation. If the expert is overloaded, many tweets can be filtered out by setting a low cost. Conversely, if the expert has time to spare, the cost can be increased to filter out normal tweets conservatively, and suspicious tweets may be referred to an expert for diagnosis.

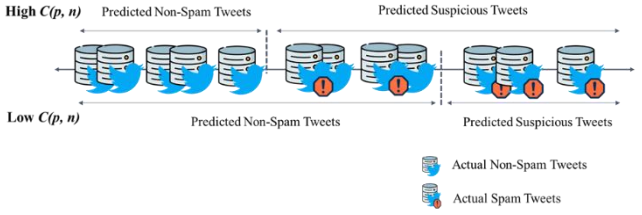

FIGURE 2. **Results of filtering according to the set cost**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098799, IEEE Access

In other words, the proposed algorithm has the advantage of detecting spam in accordance with the situation by adjusting the cost. In this study, the filtering effect and the detection accuracy were measured while varying the cost over a broad range. In addition, for cost-based machine learning classifiers, we used deep learning techniques such as deep neural networks (DNNs) along with traditional techniques such as decision tree, naïve Bayes, and RF. The machine learning algorithms and feature sets used in this study are described in Sections V.A and V.B.

C. DETECTION OF SPAM TWEETS WITH EXPERT DECISIONS

When the cost-based machine learning filter flags a tweet as suspicious, it is sent to the step where the expert makes the decision. The process by which an expert detects spam can be set in various ways, but the expert solutions of spam-detection companies [45–47] have some common characteristics. Firstly, the automatically processed spams are filtered out with priority. The typical method is the blacklist technique, which blocks URLs known as malicious in advance. Furthermore, the spams of users known as spammers are also automatically blocked. Another common characteristic is that experts respond directly to new real-time attacks.

The greatest advantage that can be obtained by the participation of experts in spam detection is reliable detection through manual inspection. It is often observed that tweets determined to be normal by the machine learning algorithm are subtle spam in the eyes of experts. With the recent increase in social engineering attacks, famous Twitter accounts are being used for attacks [48]. Furthermore, phishing attacks that send spam pretending to be acquaintances also cause constant damage. In particular, the worsening online dependence due to COVID-19 in 2020 has increased spam phishing attacks using e-mails and social media [49]. It is difficult to distinguish spam in social engineering techniques from general machine learning techniques because they are disguised as acquaintances or use the accounts of celebrities. However, experts can easily identify spam using social engineering techniques because they focus on the content of tweets, rather than whether the account belongs to a famous person or an acquaintance.

In the proposed framework, the expert inspects the tweets flagged as suspicious by the primary filter. In this situation, the tweets judged as normal by the first filter are not inspected by an expert. As mentioned above, the mixing of spam in tweets judged as normal by the first filter is prevented through cost setting. However, a 100% detection rate is practically impossible because of the nature of machine learning. Therefore, in the proposed framework, if the expert has time to spare, the process of inspecting tweets classified as normal by the primary filter is an option. This option is not often performed because it is difficult for an expert to inspect all tweets in a general situation. However, if a situation occurs in which the expert has time to spare, the tweets filtered through as normal are also inspected to improve system performance.

D. TRAINING DATA UPDATE

In the last step, the training dataset used in the filter is updated to respond to the newly evolving spam attacks. As mentioned above, most methods use only existing collected data as the training set. In this case, they have a fatal drawback in that they cannot respond to new attacks. Among the methods that attempted to improve this, we propose a technique that is appropriate for the proposed framework by developing on the method proposed in Chen et al. [42].

Classification using machine learning begins with the collection of labeled data. The spam classification is performed by an expert labeling the data after collecting existing known spam tweets and normal tweets. We denote this collected dataset T_{init}. The classification algorithm L is trained on T_{init}. The binary classifier C_{init} can be expressed as follows:

$$C_{init} = L(T_{init}).$$

Our framework also uses C_{init} to classify the labeled data collected prior. Whereas most methods use C_{init} only, we continuously update the training data. We denote N_t as the newly collected spam and normal tweets data for a predefined time interval τ. Here, t is the time unit that increases by one on the passage of the time interval τ. The definitions of the newly added dataset T_{new} and the classifier C_1 that we utilize are expressed as follows:

$$T_{new} = \bigcup_{t} N_t,$$

$$C_1 = L(T_{init} \cup T_{new}).$$

Our framework uses C_1, which has been trained with a dataset that is updated periodically, as a classifier for step 1. Therefore, it is possible to respond to the newly evolving spam tweets.

The expert classified the spam tweets and normal tweets among the data that were filtered first. However, as explained in the previous section, if the expert has time to spare, he or she can inspect whether the normal tweets filtered through by the first filter are classified properly. The dataset that receives the additional inspection of an expert is E_s, and s is the time unit that increases by one when the expert inspects filtered tweets. The updated dataset T_{new_option} and the classifier can be expressed as follows:

$$T_{new\_option} = \bigcup_{s} E_s,$$

$$C_2 = L(T_{init} \cup T_{new} \cup T_{new\_option}).$$

The performance of the filter can be maintained to obviate the latest attack trends by updating the training dataset. Furthermore, because our framework includes a process in which experts participate in decision-making, it has a system-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098799, IEEE Access

wise advantage of having no need for separate annotation for training data.

V. RESULTS AND DISCUSSION

To overcome the disadvantages of using machine learning alone, this paper proposed a sophisticated framework in which a cost-based machine learning technique and experts collaborate. A real dataset was used to verify the framework in a real environment, and an imbalanced dataset of spam tweets and normal tweets in a ratio similar to the real situation was used. Section V.A explains the dataset and evaluation metrics used in this study. Section V.B describes the performance verification results when collaborating with an expert after cost-based filtering. Section V.C verifies how the performance improves when the training dataset is updated periodically. Section V.D discusses the practicability of the proposed framework.

A. DATASET DESCRIPTION AND EVALUATION METRICS

This study used the dataset presented by Chen et al. [19] after analyzing 600 million tweets. This open dataset has been used in many studies because it comprises data collected from a real environment. Among the datasets presented by them, we used the dataset in which the ratio of spam to normal data was 5:95 for verification in a real situation. In addition, we used the "continuous" dataset to verify the update effect over time. This dataset has 12 features in total, which consist of statistical features related to tweets and account information features. The total number of tweets was 100,000. Detailed descriptions of these features are provided in Table III.

TABLE III
FEATURES DESCRIPTION

Type	Feature Name	Feature Description
Tweet-based Features	no_userfavourites	Number of favorites this Twitter user received
	no_lists	Number of lists added by the user who sent the tweet
	no_tweet	Number of posts of the user who sent the tweet
	no_retweet	Number of retweets of the tweet
	no_hashtags(#)	Number of hashtags included in the tweet
	no_usermention(@)	Number of user mentions included in the tweet
	no_url	Number of URLs included in the tweet
	no_char	Number of characters in the tweet
	no_digit	Number of digits in the tweet
Account-based Features	account_age	Number of days from the time when the account was created until the latest tweet was sent
	no_follower	Number of followers of the user who sent the tweet
	no_following	Number of followings of the user who sent the tweet

In this study, precision, recall, and F-Measure were considered as the evaluation metrics. They are calculated based on the true positive (TP), false positive (FP), and false negative (FN). Recall indicates how many actual classes were detected and is equal to the true positive rate. Precision is the accuracy of detection and refers to the probability that when a tweet is classified into a specific class, it actually falls into that class. F-Measure is the harmonic average of precision and recall, and it indicates the total performance. The equations for precision, recall, and F-Measure are as follows:

$$Precision = \frac{TP}{TP + FP},$$

$$Recall = \frac{TP}{TP + FN},$$

$$F - Measure = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}.$$

B. PERFORMANCE COMPARISON BETWEEN CONVENTIONAL METHODS AND OUR FRAMEWORK

The proposed framework is composed of three steps: a primary filter for cost-based machine learning, secondary detection based on experts, and update using the expert diagnosis result in the training data. In this section, to examine the effect of cost-based machine learning, the results of steps 1 and 2 are compared with those of the existing machine learning techniques. The effects of training data updates on new attack detection are examined in the next section.

First, we compared the performance of our proposed cost-based framework and that of the existing methods. Thus, we first classified spam and normal tweets using the existing conventional machine learning techniques such as J48, RF, PART, naïve Bayes, and MLP, which we utilized in Section III. In addition, LightGBM, which is a tree-based ensemble learning method known to have good performance and efficiency, was also used [50]. These same algorithms were used as the machine learning algorithms in the filtering step of our proposed framework. Furthermore, experiments were conducted while the cost was changed from 5 to 30 in increments of 5. The upper limit of the cost was set to 30 because the results were not significantly different when the cost was higher than 30. For algorithm implementation, Weka [51] and scikit-learn were used. Training and classification were performed through 10-fold cross-validation. Table IV presents the performance comparison for the spam tweets, and the performance results for the normal tweets are provided in the Appendix because of the voluminous data. Furthermore, the recall of spam, which is the spam-detection rate and the most important metric, is expressed separately in Fig. 3.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098799, IEEE Access

TABLE IV
COMPARISON OF PERFORMANCE FOR THE SPAM TWEETS BETWEEN SPAM FILTER AND OVERALL FRAMEWORK

ML Algorithms	Cost	Cost-based ML Filter			Overall Framework (w/Experts)		
		Precision	Recall	F-Measure	Precision	Recall	F-Measure
J48	Only ML	0.930	0.866	0.897	0.930	0.866	0.897
	5	0.896	0.879	0.887	1.000	0.879	0.935
	10	0.807	0.888	0.846	1.000	0.888	0.941
	15	0.802	0.893	0.845	1.000	0.893	0.944
	20	0.803	0.894	0.846	1.000	0.894	0.944
	25	0.798	0.894	0.844	1.000	0.894	0.944
	30	0.792	0.894	0.840	1.000	0.894	0.944
Random Forest	Only ML	0.975	0.883	0.926	0.975	0.883	0.926
	5	0.951	0.895	0.922	1.000	0.895	0.945
	10	0.912	0.903	0.908	1.000	0.903	0.949
	15	0.865	0.911	0.887	1.000	0.911	0.953
	20	0.812	0.915	0.860	1.000	0.915	0.955
	25	0.767	0.919	0.836	1.000	0.919	0.958
	30	0.724	0.923	0.812	1.000	0.923	0.960
PART	Only ML	0.923	0.841	0.880	0.923	0.841	0.880
	5	0.901	0.850	0.875	1.000	0.850	0.919
	10	0.751	0.867	0.805	1.000	0.867	0.929
	15	0.712	0.881	0.788	1.000	0.881	0.937
	20	0.591	0.901	0.714	1.000	0.901	0.948
	25	0.635	0.884	0.739	1.000	0.884	0.939
	30	0.612	0.891	0.726	1.000	0.891	0.942
Naïve Bayes	Only ML	0.603	0.717	0.655	0.603	0.717	0.655
	5	0.324	0.745	0.451	1.000	0.745	0.854
	10	0.293	0.753	0.422	1.000	0.753	0.859
	15	0.278	0.761	0.407	1.000	0.761	0.864
	20	0.267	0.763	0.396	1.000	0.763	0.866
	25	0.260	0.765	0.388	1.000	0.765	0.867
	30	0.256	0.770	0.384	1.000	0.770	0.870
MLP	Only ML	0.890	0.563	0.690	0.890	0.563	0.690
	5	0.765	0.593	0.668	1.000	0.593	0.745
	10	0.693	0.605	0.646	1.000	0.605	0.754
	15	0.625	0.618	0.622	1.000	0.618	0.764
	20	0.358	0.668	0.467	1.000	0.668	0.801
	25	0.238	0.741	0.361	1.000	0.741	0.851
	30	0.198	0.768	0.315	1.000	0.768	0.869
LightGBM	Only ML	0.971	0.886	0.927	0.971	0.886	0.927
	5	0.957	0.897	0.926	1.000	0.897	0.946
	10	0.914	0.906	0.910	1.000	0.906	0.951
	15	0.863	0.913	0.888	1.000	0.913	0.955
	20	0.803	0.921	0.858	1.000	0.921	0.959
	25	0.760	0.925	0.835	1.000	0.925	0.961
	30	0.725	0.928	0.814	1.000	0.928	0.963

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098799, IEEE Access

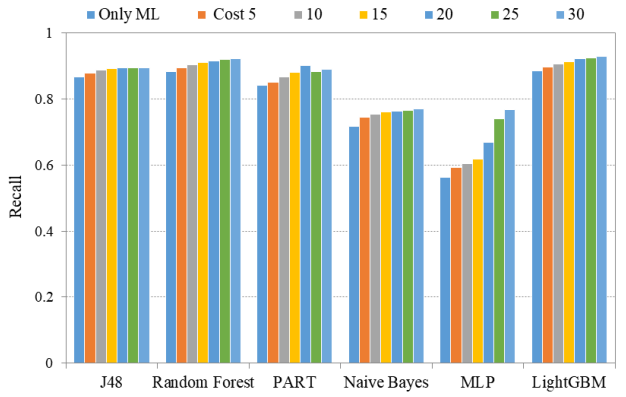

FIGURE 3. **Spam recall according to cost change for each machine learning algorithm**

The cost-based machine learning filter in Table IV shows the results of step 1 of the proposed framework. The overall framework shows the results of step 2 (expert diagnosis). "Only ML" refers to applying only the conventional machine learning algorithm. The existing machine learning algorithm cannot be divided into steps 1 and 2 because the machine learning algorithm is used for classification, but it is marked with duplicates to compare the performance with that of our framework. As our framework allows the expert to diagnose tweets that were not filtered by the primary filter, it is crucial to prevent the primary filter from classifying spam tweets as normal. For each algorithm, as the cost increases, the recall, which is the most important parameter, also increases in varying degrees. It can be seen that even with the lowest cost, our framework filters out more spam tweets than the conventional machine learning techniques. The performance improved from approximately a minimum of 3% to a maximum of approximately 20%. This change in the recall trend can also be seen in Fig. 3. The highest performance was observed when LightGBM and cost were used together, filtering out up to 92.8% of spam. In other words, using cost-based machine learning as the primary filter is more effective than using simple machine learning as the primary filter. The precision is relatively low for the results of the primary filter alone. This is because even slightly suspicious tweets are sent to the second step. Because the expert inspection in the second step can perfectly detect normal tweets, the results of the overall framework are considerably high.

Fig. 4 shows the number of undetected spam tweets when a cost is applied and when no cost is applied for each machine learning algorithm. LightGBM, which exhibits the highest performance, did not detect 571 spam tweets. By contrast, when our framework was applied, it detected up to 212 more spam tweets. J48, RF, and PART with the proposed framework also detected approximately 200 more spam tweets compared with the case of using them alone; MLP with the cost detected more than 1,000 spam tweets. However, MLP and naïve Bayes could not detect many spam tweets even

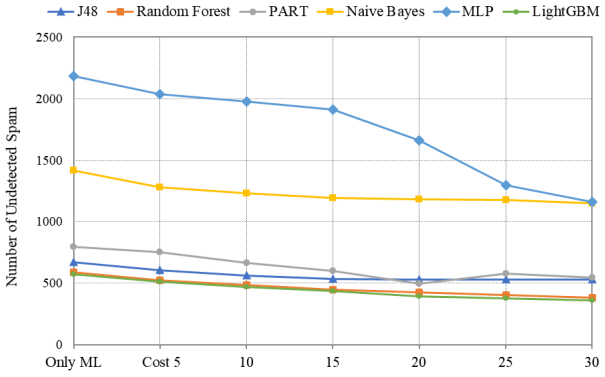

FIGURE 4. **Number of undetected spam tweets for various costs.**

when our framework was applied because their detection performance was poor.

The above results confirm that spam tweets can be detected better if the cost is higher. However, if the cost is high, even slightly suspicious tweets are sent to the expert, which increases the workload of the expert. Fig. 5 shows the change in the load on the experts as the cost increases. The y-axis in Fig. 5 shows the ratio of the tweets inspected by experts to all the tweets. In the case of J48, RF, PART, and LightGBM, which show high performance, the higher the cost, the higher the ratio of tweets inspected by experts. However, it can be seen that only 5% to 10% of all tweets should be inspected by experts. In the case of MLP and naïve Bayes, which do not have good performance, as the cost increases, the load on experts becomes considerably high. This suggests that by using a machine learning algorithm with good performance along with a cost-based filter, a good spam-detection rate can be achieved, and the load on experts can be considerably reduced.

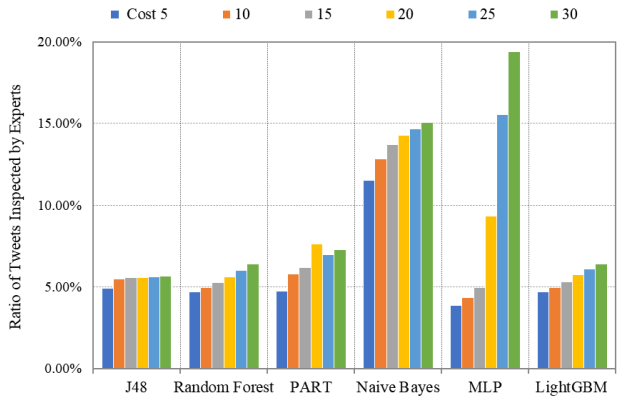

FIGURE 5. Ratio of tweets inspected by experts according to cost change.

C. PERFORMANCE COMPARISON WITH THE EVOLUTION OF SPAM ATTACKS

In the previous sections, training and classification were performed under the assumption that all data were collected in advance. This assumption is far from reality, where new spam

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098799, IEEE Access

types appear continuously. Therefore, we propose a module that updates the training data used in the primary filter during regular periods based on the results of expert diagnosis. To compare the performance whenever the spam attack evolves, we used the "95k-continuous" dataset, which is a collection of data in the order of time, created by Chen et al. [19]. If a dataset of 100,000 data points in total is divided into 10 datasets, and each dataset is assumed to be data collected for a day, it can be classified into datasets collected from day 1 to day 10. Firstly, for comparison with the conventional machine learning, the performance of the spam-detection method was measured when the data collected on day 1 were used as the training data, and the data collected on days 2 to 10 were used as the test data. For evaluating our framework, after updating the training data used in the filter (step 1) with the data classification result of the previous day, the performance of the overall framework was measured. The cost of the step 1 filter was set to 15, which showed balanced results. Furthermore, among the five machine learning algorithms used in the previous section, J48, RF, PART, and LightGBM, which showed excellent performance, were used here. Fig. 6 shows the change in the recall for spam tweets each time the date is changed, and Fig. 7 shows the changes in F-Measure for spam tweets each time the date is changed.

Until day 2, our method and the general machine learning technique showed almost no difference in performance. However, from day 3, the values of recall began to show differences. Particularly on days 4, 6, and 9, the recall of the general machine learning techniques decreased by approximately 0.5. This means that half of all spam tweets were not detected. It can be seen that the recall values are significantly different from when the entire dataset was used for training and classification in the previous section. In contrast, the proposed framework shows consistent accuracy even though the performance changes slightly because it updates the data by reflecting the results of classification by experts. In particular, the performance is high even on days when the recall of the general machine learning technique is much lower. This suggests that our algorithm can also respond well to changes in spam over time. Furthermore, the proposed algorithm shows consistent values for the F-Measure, which indicates the overall performance, whereas the general machine learning techniques show large deviations according to the date.

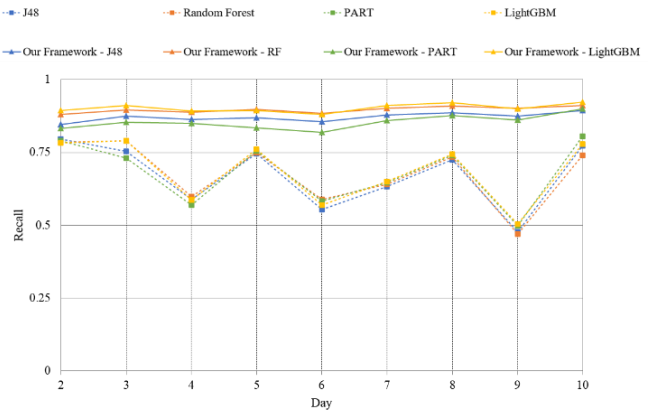

FIGURE 6. **Recall for spam tweets over time.**

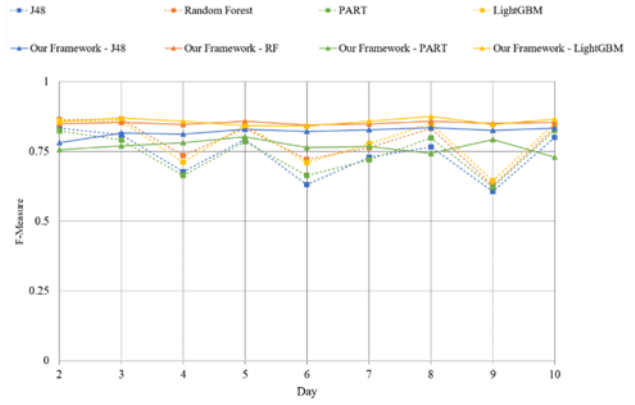

FIGURE 7. **F-Measure for spam tweets over time.**

D. DISCUSSION

In this study, we proposed a framework for experts and machine learning techniques to collaborate in detecting spam on social networks. As mentioned above, several studies have used artificial intelligence techniques, such as machine learning, to detect spam tweets. These studies focused on improving the spam detection performance based on public datasets or collected data. However, we have shown by experiments that it is difficult to detect spam using only machine learning because of various problems encountered in reality. Therefore, we proposed a collaborative system including machine learning and experts for spam detection while reducing the time-consuming or costly expenses that can be problematic when experts are involved. Our framework was able to improve performance compared with that of the existing methods, and it has the advantage of being able to adjust the degree of expert participation through cost adjustment. Furthermore, our proposed framework is significant in that it is the first of its kind in the field of spam tweets detection, to our knowledge.

However, as this is the first proposed method, questions may be raised in terms of practicability. As can be seen from

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098799, IEEE Access

[45-47] discussed above, experts often participate in commercial spam detection programs, meaning our technique is feasible. To further verify the practicability of our technique, we explored various studies involving a collaboration with experts in other security fields.

First, we reviewed studies on solving the problem of excessive cost spending and time-consumption when experts participate in the security system. In order to adjust the degree of expert participation, a method of classifying security threats into grades according to severity was proposed. Using graded security in an expert system achieves a fast security system with sufficiently high confidence [52]. In a similar study, knowledge bases were constructed with the help of experts. Because expert participation is expensive, a method of varying the degree of expert intervention according to problem area, threat level, and security asset has been proposed [53]. The cyber assets planning process, which is an important element in configuring a security system, also depends considerably on experts, and this process is time-consuming and sometimes inefficient. To address these shortcomings, Costa et al. proposed an approach to streamline the cyber assets planning process by using probabilistic ontologies [54]. Analyzing the security log also requires expert participation, which is often difficult or time-consuming for non-experts. Therefore, there is a need for a system that can be easily used by experts who have little security knowledge. Khan and Parkinson extracted domain action models from event logs using rule mining, and by this, non-experts could perform expert analysis [55].

Studies have also been conducted to solve problems that may arise when various experts participate. In order to effectively respond to cyber security risks, an approach was suggested for experts in different fields to easily collaborate [56]. Additionally, a study was conducted on the reflection of expert opinions when the opinions are different, and the authors stated that it is important to match the consensus among experts [57]. Other studies involving a collaboration with experts in detecting specific attacks have also been conducted. To prevent a distributed denial of service attack in the IoT environment, a method using software-defined networking with the help of experts has been proposed [58]. Additionally, a method to reflect expert feedback in anomaly detection techniques to detect new types of cyber attacks has also been proposed [59].

Consequently, several studies have been conducted in which experts and security systems collaborate. In particular, many studies have considered different degrees of expert participation in solving problems that may arise when involving experts. This can be seen as a similar approach to our method as our proposed technique considers different degrees of expert intervention through filtering using machine learning. That is, as proven in several previous studies, our proposed collaboration system is also sufficiently practicable. In fact, when experts are involved in detecting spam, our framework enables the reduction of the load on the experts and maintains the detection performance above a certain level.

VI. CONCLUSION

This paper proposed a sophisticated framework in which experts and machine learning algorithms collaborate to detect spam tweets effectively. As many normal tweets could be filtered out through the cost-based machine learning filter in the first step, it can reduce the workload on the expert. The accuracy can also be improved by subjecting the suspicious tweets to analysis by experts. The experimental results show that the proposed method has a higher spam-detection rate than the conventional machine learning technique. Furthermore, a module that periodically updates the training dataset is constructed to reflect the trend of the tweets that change continuously. The conventional machine learning technique does not respond to the change in trends over time, and it shows inconsistent performance. However, our framework showed consistently higher performance than that of the existing technique through periodic updates.

This study contributes to the spam-detection field as follows. Firstly, a framework suitable for practical utilization is proposed. Most existing methods use datasets composed of normal and spam data in a 1:1 ratio, which is far from reality. Thus, they have limitations in that they do not reflect reality although they guarantee high performance. In contrast, our framework is trained using datasets that reflect reality with a normal to spam tweet ratio of 95:5, and it was found to maintain high performance through the collaboration of experts and machine learning, even in a real-world situation. This has implications for detecting spam in real environments in the future.

Secondly, the proposed framework can respond even to evolving spam attacks. Machine learning algorithms that are trained on the collected data cannot respond effectively to changing trends in spam data, even though their performance for the existing data is excellent. Our framework periodically updates the dataset with the diagnosis results of experts in the primary filter. Thus, the filter itself shows good performance for evolving spam attacks.

Thirdly, the proposed framework can be flexibly operated in accordance with the situation. If the cost of the primary filter is high, the detection rate for spam can be increased. In contrast, if the cost is low, the load on the experts can be reduced. This property can be used to reduce the load on experts by lowering the cost if there are many tweets to be inspected, or to improve performance by raising the cost if the load on experts is low. The advantage of our framework is that it is suitable for use in realistic scenarios through cost setting in accordance with the situation.

In future studies, methods that can improve the performance of the proposed method should be considered. Although the proposed framework has achieved good performance, additional research on using deep learning can be considered to improve the detection rate further. Furthermore, the dataset used in this study and many other spam-detection studies was released in 2015. Thus, the dataset does not reflect the current trends. If a reliable dataset that reflects the current reality is

made available in the future, further research based on it should be conducted.

APPENDIX

The performance results for general machine learning and the cost-based filter for normal tweets are provided in Table A.1.

REFERENCES

[1] Internet Usage Statistics. https://internetworldstats.com/stats.htm/, 2020 (accessed 14 January 2021).

[2] Statistica, "Number of smartphone users from 2016 to 2021," https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/, 2020 (accessed 14 January 2021).

[3] A. Ali, "Visualizing the social media universe in 2020," https://www.visualcapitalist.com/visualizing-the-social-media-universe-in-2020/, 2020 (accessed 14 January 2021).

[4] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," Commun. ACM 59(7) (2016) 96-104. doi:10.1145/2818717

[5] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," Science 359(6380) (2018) 1146-1151.

[6] N. H. Imam and V. G. Vassilakis, "A survey of attacks against Twitter spam detectors in an adversarial environment," Robotics, 8(3) (2019) 50. doi:10.3390/robotics8030050

[7] O. Varol, E. Ferrara, C. B. Davis, F. Menczer, and A. Flammini, "Online Human-Bot Interactions: Detection, Estimation, and Characterization," In: Proceedings of the Eleventh International AAAI Conference on Web and Social Media (ICWSM 2017), 2017.

[8] C. Grier, K. Thomas, V. Paxson, and M. Zhang, "@ spam: The underground on 140 characters or less," In: Proceedings of the 17th ACM conference on Computer and communications security, 2010.

[9] Reuters, 'Twitter suspends accounts claiming to be Black Trump supporters over spam manipulation," Reuters. https://www.reuters.com/article/us-usa-twitter-disinformation/twitter-suspends-accounts-claiming-to-be-black-trump-supporters-over-spam-manipulation-idUSKBN26Y2ZM/, 2020 (accessed 14 January 2021).

[10] R. Lerman and H. Denham, "3 charged in massive Twitter hack, including alleged teenage 'mastermind'," The Washington Post. https://www.washingtonpost.com/technology/2020/07/30/twitter-hack-phone-attack/, 2020 (accessed 14 January 2021).

[11] K. Lee, B. D. Eoff, and J. Caverlee, "Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter," In: International AAAI Conference on Web and Social Media (ICWSM), 2011.

[12] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," In: Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC), 2010.

[13] G. Wang, M. Mohanlal, C. Wilson, X. Wang, M. Metzger, H. Zheng, B. and Y. Zhao, "Social Turing Tests: Crowdsourcing Sybil Detection," In: NDSS Symposium 2013.

[14] 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 Secur. 12(4) (2017) 914-925. doi:10.1109/tifs.2016.2621888

[15] M. Washha, A. Qaroush, M. Mezghani, and F. Sedes, "Unsupervised collective-based framework for dynamic retraining of supervised real-time spam tweets detection model," Expert Syst. Appl. 135 (2019) 129-152. doi:10.1016/j.eswa.2019.05.052

[16] T. Wu, S. Wen, Y. Xiang, and W. Zhou, "Twitter spam detection: Survey of new approaches and comparative study," Comput. Secur. 76 (2017) 265-284. doi:10.1016/j.cose.2017.11.013

[17] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on Twitter," In: Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS), 2010.

[18] C. Chen, J. Zhang, Y. Xie, Y. Xiang, W. Zhou, M. M. Hassan, A. AlElaiwi, and M. Alrubaian, "A performance evaluation of machine learning-based streaming spam tweets detection," IEEE Trans. Comput. Social Syst. 2(3) (2015) 65-76.

[19] C. Chen, J. Zhang, X. Chen, Y. Xiang, and W. Zhou, "6 Million Spam Tweets: A Large Ground Truth for Timely Twitter Spam Detection," In: 2015 IEEE International Conference on Communications (ICC), 2015.

[20] F. Ahmed and M. Abulaish, "A generic statistical approach for spam detection in Online Social Networks," Comput. Commun. 36(10-11) (2013) 1120-1129. doi:10.1016/j.comcom.2013.04.004

[21] S. Liu, J. Zhang, and Y. Xiang, "Statistical detection of online drifting twitter spam," In: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, 2016.

[22] A. Aizawa, "The feature quantity: An information theoretic perspective of tfidf-like measures," In: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, 2000.

[23] Z. Chu, I. Widjaja, and H. Wang, "Detecting social spam campaigns on twitter," In: International Conference on Applied Cryptography and Network Security, 2012.

[24] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu, "Analyzing spammers' social networks for fun and profit: A case study of cyber criminal ecosystem on twitter," In: Proceedings of the 21st international conference on World Wide Web, 2012.

[25] X. Hu, J. Tang, and H. Liu, "Online social spammer detection," In: Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI-14), 2014.

[26] I. Inuwa-Dutse, M. Liptrott, and I. Korkontzelos, "Detection of spam-posting accounts on Twitter," Neurocomputing, 315 (2018) 496-511. doi:10.1016/j.neucom.2018.07.044

[27] S. Kudugunta and E. Ferrara, "Deep neural networks for bot detection," Inf. Sci. 467 (2018) 312-322. doi:10.1016/j.ins.2018.08.019

[28] S. Madisetty and M. S. Desarkar, "A Neural Network-Based Ensemble Approach for Spam Detection in Twitter," IEEE Trans. Comput. Social Syst. 5(4) (2018) 973-984. doi:10.1109/tcss.2018.2878852

[29] S. Sedhai and A. Sun, "An Analysis of 14 Million Tweets on Hashtag-Oriented Spamming," J. Assoc. Inf. Sci. Technol. 68(7) (2017a) 1638-1651. doi:10.1002/jasist.23836

[30] S. Sedhai and A. Sun, "Semi-Supervised Spam Detection in Twitter Stream," IEEE Trans. Comput. Social Syst. 5(1) (2017b) 169-175.

[31] H. Costa, F. Benevenuto, and L. H. Merschmann, "Detecting tip spam in location-based social networks," In: Proceedings of the 28th International Symposium on Applied Computing (SAC 2013), 2013.

[32] X. Hu, J. Tang, Y. Zhang, and H. Liu, "Social spammer detection in microblogging," In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, 2013.

[33] C. Yang, R. Harkreader, and G. Gu, "Empirical evaluation and new design for fighting evolving twitter spammers," IEEE Trans. Inf. Forensics Secur. 8(8) (2013) 1280-1293.

[34] C. Chen, J. Zhang, Y. Xiang, W. Zhou, and J. Oliver, "Spammers are becoming "Smarter" on Twitter," IT Prof. 18(2) (2016) 66-70.

[35] A. H. Wang, "Don't follow me: Spam detection in Twitter," In: International Conference on Security and Cryptography (SECRYPT 2010), 2010.

[36] J. Parsons, "Is It Legal to Buy and Sell Twitter Accounts?" https://follows.com/blog/2016/02/legal-buy-sell-twitter-accounts, 2016 (accessed 14 January 2021).

[37] C. Li and S. Liu, "A comparative study of the class imbalance problem in Twitter spam detection," Concurrency Comput.: Pract. and Experience, 30(5) (2018) e4281. doi:10.1002/cpe.4281

[38] Q. Yang and X. Wu, "10 challenging problems in data mining research," Int. J. Inf. Technol. Decis. Making, 5(04) (2006) 597-604.

[39] S. Liu, Y. Wang, J. Zhang, C. Chen, and Y. Xiang, "Addressing the class imbalance problem in twitter spam detection using ensemble learning," Comput. Secur. 69 (2017) 35-49.

[40] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res. 16 (2002) 321-357.

[41] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Trans. Knowl. Data Eng. 21(9) (2009) 1263-1284.

[42] C. Chen, J. Zhang, Y. Xiang, and W. Zhou, "Asymmetric self-learning for tackling twitter spam drift,' In: 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2015.

[43] P. Domingos, "Metacost: A general method for making classifiers cost-sensitive," In: Proceedings of the fifth ACM SIGKDD

international conference on Knowledge discovery and data mining, 1999.

[44] C. Zhao, Y. Xin, X. Li, Y. Yang, and Y. Chen, "A Heterogeneous Ensemble Learning Framework for Spam Detection in Social Networks with Imbalanced Data," Appl. Sci. 10(3) (2020) 936. doi:10.3390/app10030936

[45] WHUK. https://www.webhosting.uk.com/, 2020 (accessed 14 January 2021).

[46] eukhost. https://www.eukhost.com/email-spam-filter/, 2020 (accessed 14 January 2021).

[47] bluehost. https://www.bluehost.com/help/article/spamexperts/, 2020 (accessed 14 January 2021).

[48] L. Cohen, "Twitter says hacking of high-profile Twitter accounts was a "coordinated social engineering attack"." CBS News. https://www.cbsnews.com/news/twitter-hack-verified-accounts-social-engineering-bitcoin-scam/, 2020 (accessed 14 January 2021).

[49] J. Kobielus, "Social engineering hacks weaken cybersecurity during the pandemic," InfoWorld. https://www.infoworld.com/article/3565197/social-engineering-hacks-weaken-cybersecurity-during-the-pandemic.html/, 2020 (accessed 14 January 2021).

[50] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu, "LightGBM: A highly efficient gradient boosting decision tree," Adv. Neural Inf. Process. Syst. 30 (2017) 3146-3154.

[51] F. Eibe, M. A. Hall, and I. H. Witten, *The WEKA workbench. Online appendix for data mining: Practical machine learning tools and techniques*, Morgan Kaufmann, 2016.

[52] J. Kivimaa, A. Ojamaa, and E. Tyugu, "Graded security expert system," in International Workshop on Critical Information Infrastructures Security, 2008: Springer, pp. 279-286.

[53] L. Atymtayeva, K. Kozhakhmet, and G. Bortsova, "Building a knowledge base for expert system in information security," in Soft Computing in Artificial Intelligence: Springer, 2014, pp. 57-76.

[54] P. C. Costa, B. Yu, M. Atiahetchi, and D. Myers, "High-level information fusion of cyber-security expert knowledge and experimental data," in 2018 21st International Conference on Information Fusion (FUSION), 2018: IEEE, pp. 2322-2329.

[55] S. Khan and S. Parkinson, "Discovering and utilising expert knowledge from security event logs," J. Inf. Secur. Appl. 48 (2019). doi: 10.1016/j.jisa.2019.102375.

[56] M. G. Cains, L. Flora, D. Taber, Z. King, and D. S. Henshel, "Defining cyber security and cyber security risk within a multidisciplinary context using expert elicitation," Risk Anal. 2021.

[57] H. Holm, T. Sommestad, M. Ekstedt, and N. Honeth, "Indicators of expert judgement and their significance: an empirical investigation in the area of cyber security," Expert Syst. 31(4) (2014) 299-318. doi: 10.1111/exsy.12039.

[58] A. Mubarakali, K. Srinivasan, R. Mukhalid, S. C. Jaganathan, and N. Marina, "Security challenges in internet of things: Distributed denial of service attack detection using support vector machine-based expert systems," Comput. Intell. 36(4) (2020) 1580-1592.

[59] M. A. Siddiqui, J. W. Stokes, C. Seifert, E. Argyle, R. McCann, J. Neil, and J. Carroll, "Detecting cyber attacks using anomaly detection with explanations and expert feedback," in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019: IEEE, pp. 2872-2876.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098799, IEEE Access

TABLE A.1
COMPARISON OF PERFORMANCE FOR THE NORMAL TWEETS BETWEEN SPAM FILTER AND OVERALL FRAMEWORK

ML Algorithms	Cost	Cost-based ML Filter			Overall Framework (w/Experts)		
		Precision	Recall	F-Measure	Precision	Recall	F-Measure
J48	Only ML	0.993	0.997	0.995	0.993	0.997	0.995
	5	0.994	0.995	0.994	0.994	1.000	0.997
	10	0.994	0.989	0.991	0.994	1.000	0.997
	15	0.994	0.988	0.991	0.994	1.000	0.997
	20	0.994	0.988	0.991	0.994	1.000	0.997
	25	0.994	0.988	0.991	0.994	1.000	0.997
	30	0.994	0.988	0.991	0.994	1.000	0.997
Random Forest	Only ML	0.994	0.999	0.996	0.994	0.999	0.996
	5	0.994	0.998	0.996	0.995	1.000	0.997
	10	0.995	0.995	0.995	0.995	1.000	0.997
	15	0.995	0.993	0.994	0.995	1.000	0.998
	20	0.995	0.989	0.992	0.996	1.000	0.998
	25	0.996	0.985	0.990	0.996	1.000	0.998
	30	0.996	0.981	0.989	0.996	1.000	0.998
PART	Only ML	0.992	0.996	0.994	0.992	0.996	0.994
	5	0.992	0.995	0.994	0.992	1.000	0.996
	10	0.993	0.985	0.989	0.993	1.000	0.997
	15	0.994	0.981	0.987	0.994	1.000	0.997
	20	0.995	0.967	0.981	0.995	1.000	0.997
	25	0.994	0.973	0.983	0.994	1.000	0.997
	30	0.994	0.970	0.982	0.994	1.000	0.997
Naïve Bayes	Only ML	0.985	0.975	0.980	0.985	0.975	0.980
	5	0.986	0.918	0.951	0.987	1.000	0.993
	10	0.986	0.904	0.943	0.987	1.000	0.994
	15	0.986	0.896	0.939	0.988	1.000	0.994
	20	0.986	0.890	0.936	0.988	1.000	0.994
	25	0.986	0.886	0.933	0.988	1.000	0.994
	30	0.986	0.882	0.931	0.988	1.000	0.994
MLP	Only ML	0.977	0.996	0.987	0.977	0.996	0.987
	5	0.979	0.990	0.985	0.979	1.000	0.989
	10	0.979	0.986	0.983	0.980	1.000	0.990
	15	0.980	0.981	0.980	0.980	1.000	0.990
	20	0.982	0.937	0.959	0.983	1.000	0.991
	25	0.985	0.876	0.927	0.987	1.000	0.993
	30	0.986	0.836	0.905	0.988	0.990	0.990
LightGBM	Only ML	0.994	0.999	0.996	0.994	0.999	0.996
	5	0.995	0.998	0.996	0.995	1.000	0.997
	10	0.995	0.996	0.995	0.995	1.000	0.998
	15	0.995	0.992	0.994	0.995	1.000	0.998
	20	0.996	0.988	0.992	0.996	1.000	0.998
	25	0.996	0.985	0.990	0.996	1.000	0.998
	30	0.996	0.982	0.989	0.996	1.000	0.998

TABLES

TABLE I
COMPARISON OF THE PERFORMANCE OF VARIOUS MACHINE LEARNING ALGORITHMS FOR DATASETS 1 AND 2

Machine Learning Algorithms		Dataset 1 (Normal:Spam = 1:1)			Dataset 2 (Normal:Spam = 95:5)		
		Precision	Recall	F-Measure	Precision	Recall	F-Measure
J48	Spam	0.942	0.925	0.934	0.930	**0.866**	0.897
	Normal	0.927	0.943	0.935	0.993	0.997	0.995
	Weighted Avg.	0.935	0.934	0.934	0.990	0.990	0.990
Random Forest	Spam	0.976	0.937	0.956	0.975	**0.883**	0.926
	Normal	0.940	0.977	0.958	0.994	0.999	0.996
	Weighted Avg.	0.958	0.957	0.957	0.993	0.993	0.993
PART	Spam	0.938	0.921	0.989	0.923	**0.841**	0.880
	Normal	0.922	0.939	0.930	0.992	0.996	0.994
	Weighted Avg.	0.930	0.930	0.930	0.988	0.989	0.988
Naïve Bayes	Spam	0.921	0.834	0.875	0.603	**0.717**	0.644
	Normal	0.848	0.928	0.887	0.985	0.975	0.980
	Weighted Avg.	0.885	0.881	0.881	0.966	0.962	0.964
MLP	Spam	0.900	0.870	0.885	0.890	**0.563**	0.690
	Normal	0.875	0.903	0.889	0.977	0.996	0.987
	Weighted Avg.	0.887	0.887	0.887	0.973	0.975	0.972

TABLE II
ASYMMETRIC MISCLASSIFICATION COST MATRIX

	Actual Positive	**Actual Negative**
Predicted Positive	$C(p,p)$	$C(n,p)$
Predicted Negative	$C(p,n)$	$C(n,n)$

TABLE III
FEATURES DESCRIPTION

Type	Feature Name	Feature Description
Tweet-based Features	no_userfavourites	Number of favorites this Twitter user received
	no_lists	Number of lists added by the user who sent the tweet
	no_tweet	Number of posts of the user who sent the tweet
	no_retweet	Number of retweets of the tweet
	no_hashtags(#)	Number of hashtags included in the tweet
	no_usermention(@)	Number of user mentions included in the tweet
	no_url	Number of URLs included in the tweet
	no_char	Number of characters in the tweet
	no_digit	Number of digits in the tweet
Account-based Features	account_age	Number of days from the time when the account was created until the latest tweet was sent
	no_follower	Number of followers of the user who sent the tweet
	no_following	Number of followings of the user who sent the tweet

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098799, IEEE Access

TABLE IV
COMPARISON OF PERFORMANCE FOR THE SPAM TWEETS BETWEEN SPAM FILTER AND OVERALL FRAMEWORK

ML Algorithms	Cost	Cost-based ML Filter			Overall Framework (w/Experts)		
		Precision	Recall	F-Measure	Precision	Recall	F-Measure
J48	Only ML	0.930	0.866	0.897	0.930	0.866	0.897
	5	0.896	0.879	0.887	1.000	0.879	0.935
	10	0.807	0.888	0.846	1.000	0.888	0.941
	15	0.802	0.893	0.845	1.000	0.893	0.944
	20	0.803	0.894	0.846	1.000	0.894	0.944
	25	0.798	0.894	0.844	1.000	0.894	0.944
	30	0.792	0.894	0.840	1.000	0.894	0.944
Random Forest	Only ML	0.975	0.883	0.926	0.975	0.883	0.926
	5	0.951	0.895	0.922	1.000	0.895	0.945
	10	0.912	0.903	0.908	1.000	0.903	0.949
	15	0.865	0.911	0.887	1.000	0.911	0.953
	20	0.812	0.915	0.860	1.000	0.915	0.955
	25	0.767	0.919	0.836	1.000	0.919	0.958
	30	0.724	0.923	0.812	1.000	0.923	0.960
PART	Only ML	0.923	0.841	0.880	0.923	0.841	0.880
	5	0.901	0.850	0.875	1.000	0.850	0.919
	10	0.751	0.867	0.805	1.000	0.867	0.929
	15	0.712	0.881	0.788	1.000	0.881	0.937
	20	0.591	0.901	0.714	1.000	0.901	0.948
	25	0.635	0.884	0.739	1.000	0.884	0.939
	30	0.612	0.891	0.726	1.000	0.891	0.942
Naïve Bayes	Only ML	0.603	0.717	0.655	0.603	0.717	0.655
	5	0.324	0.745	0.451	1.000	0.745	0.854
	10	0.293	0.753	0.422	1.000	0.753	0.859
	15	0.278	0.761	0.407	1.000	0.761	0.864
	20	0.267	0.763	0.396	1.000	0.763	0.866
	25	0.260	0.765	0.388	1.000	0.765	0.867
	30	0.256	0.770	0.384	1.000	0.770	0.870
MLP	Only ML	0.890	0.563	0.690	0.890	0.563	0.690
	5	0.765	0.593	0.668	1.000	0.593	0.745
	10	0.693	0.605	0.646	1.000	0.605	0.754
	15	0.625	0.618	0.622	1.000	0.618	0.764
	20	0.358	0.668	0.467	1.000	0.668	0.801
	25	0.238	0.741	0.361	1.000	0.741	0.851
	30	0.198	0.768	0.315	1.000	0.768	0.869
LightGBM	Only ML	0.971	0.886	0.927	0.971	0.886	0.927
	5	0.957	0.897	0.926	1.000	0.897	0.946
	10	0.914	0.906	0.910	1.000	0.906	0.951
	15	0.863	0.913	0.888	1.000	0.913	0.955
	20	0.803	0.921	0.858	1.000	0.921	0.959
	25	0.760	0.925	0.835	1.000	0.925	0.961
	30	0.725	0.928	0.814	1.000	0.928	0.963

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098799, IEEE Access

FIGURES

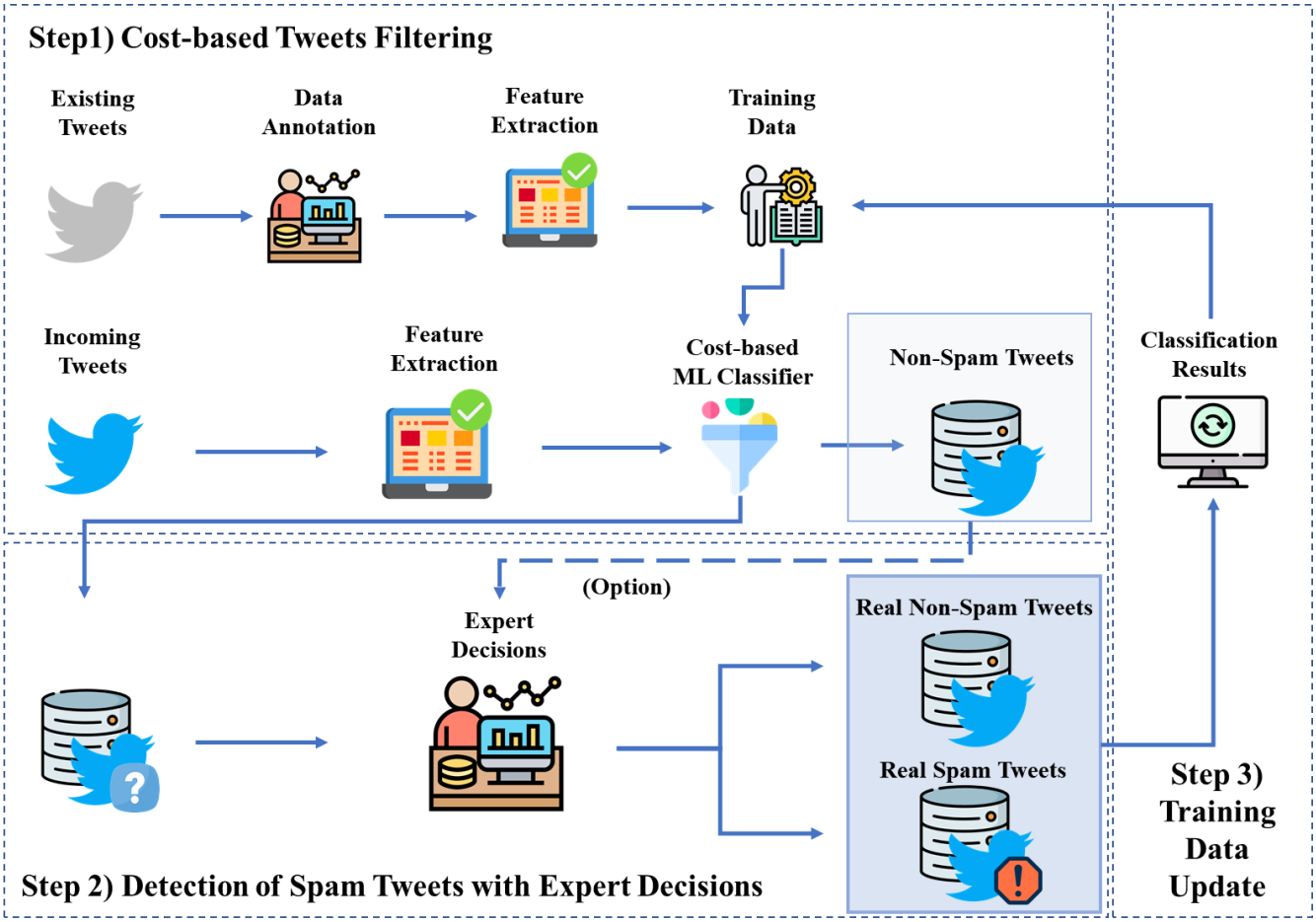

FIGURE 1. **Workflow of the proposed framework**

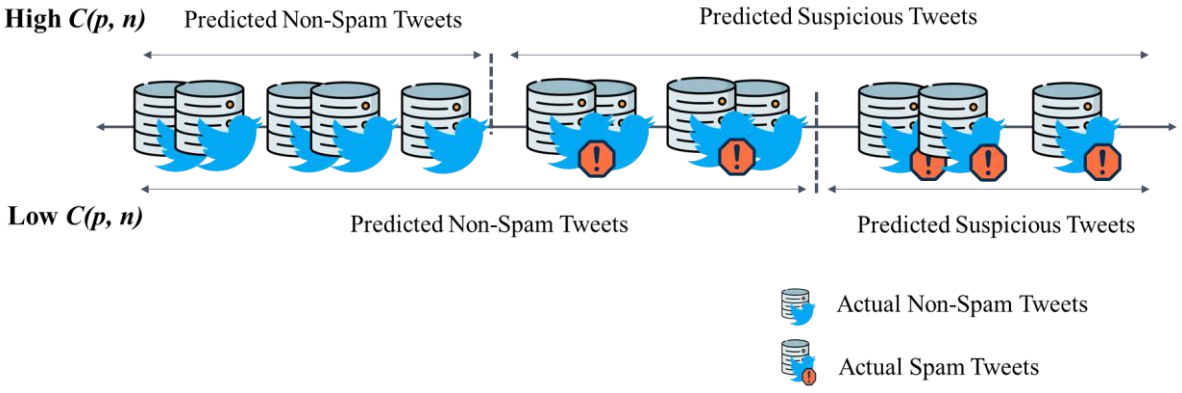

FIGURE 2. **Results of filtering according to the set cost.**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098799, IEEE Access

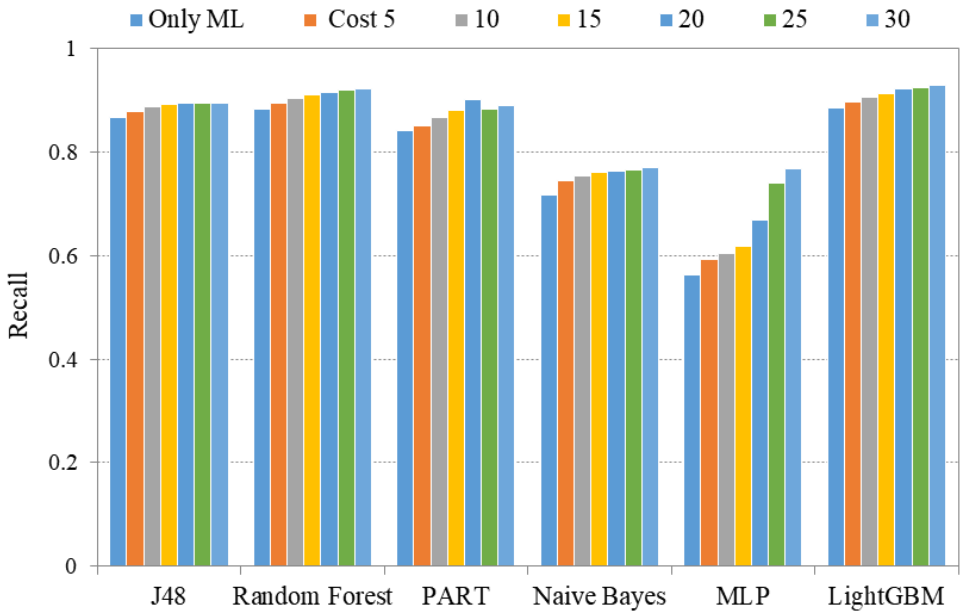

FIGURE 3. **Spam recall according to cost change for each machine learning algorithm**

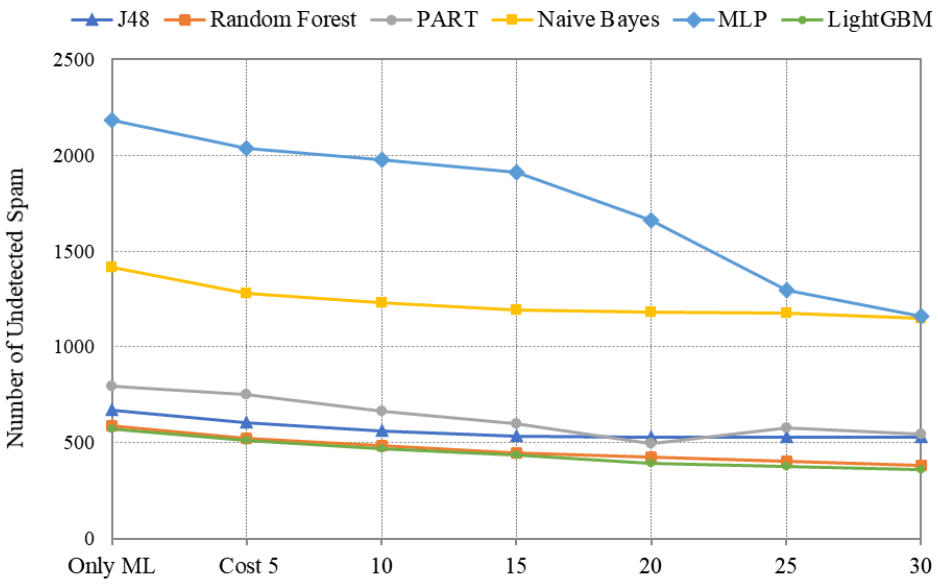

FIGURE 4. **Number of undetected spam tweets for various costs.**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098799, IEEE Access

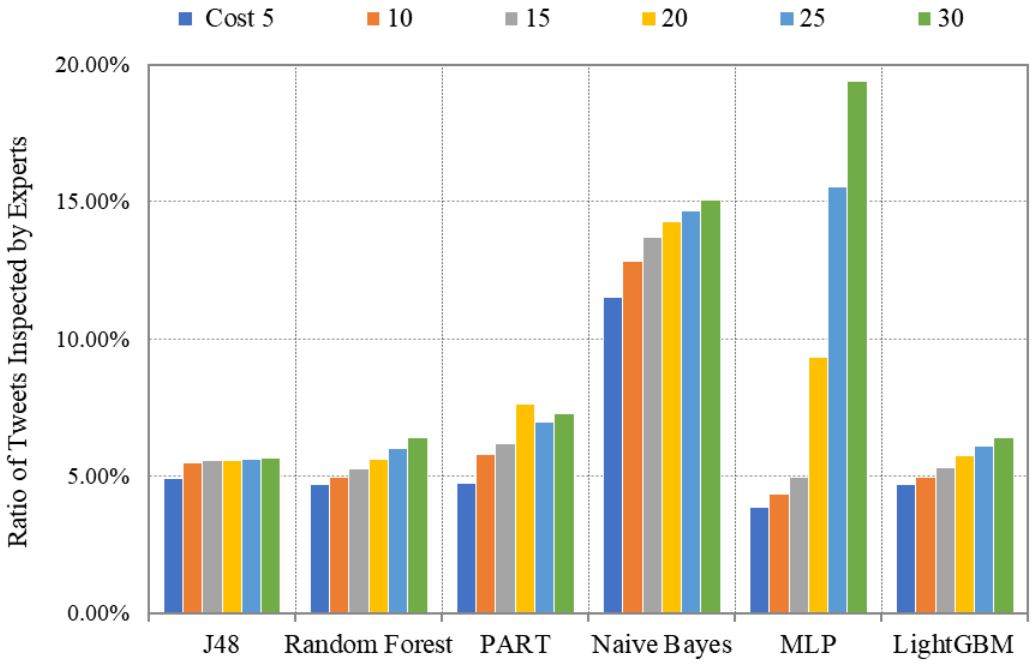

FIGURE 5. **Ratio of tweets inspected by experts according to cost change.**

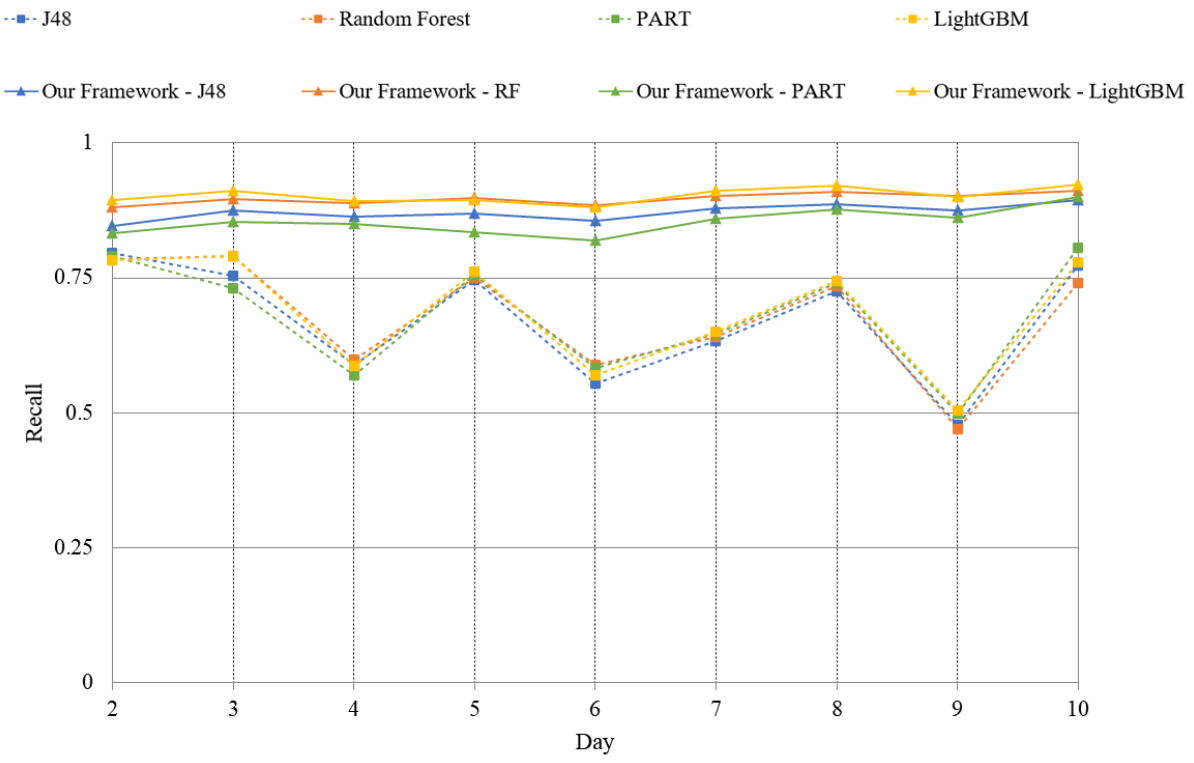

FIGURE 6. **Recall for spam tweets over time.**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3098799, IEEE Access

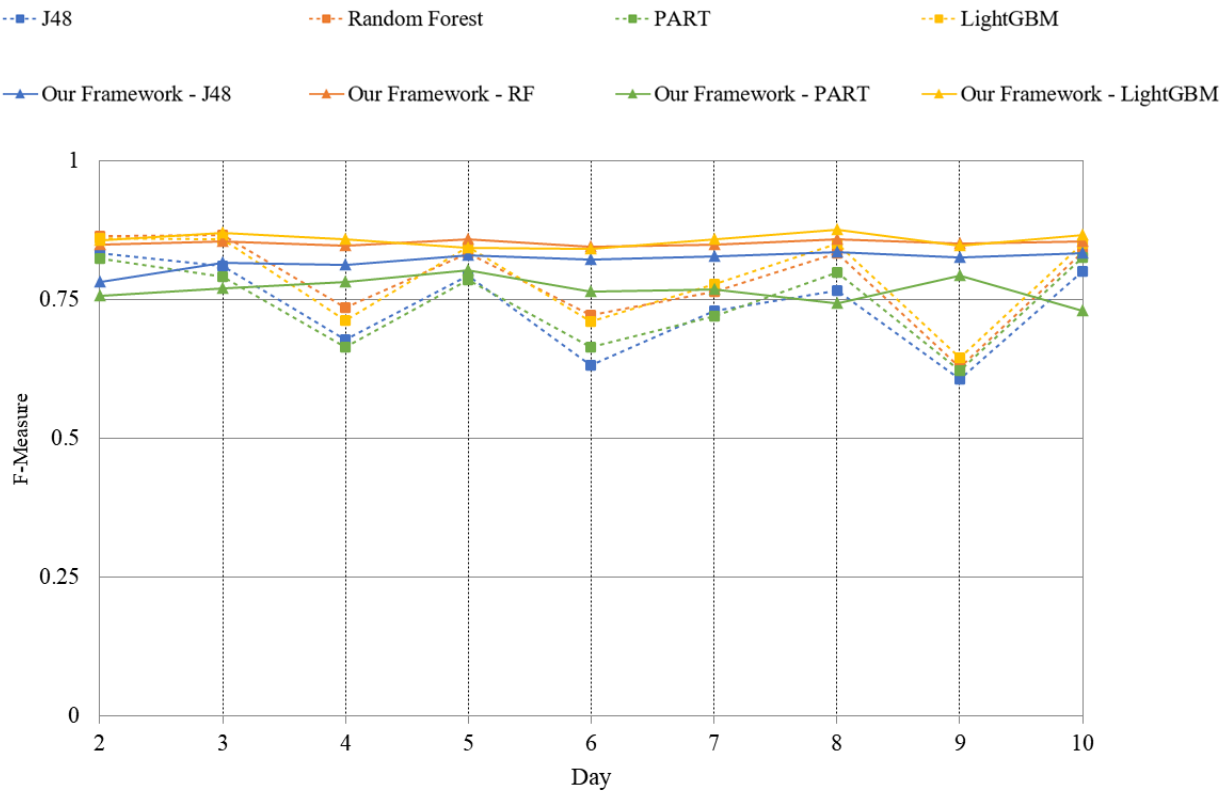

FIGURE 7. **F-Measure for spam tweets over time.**