IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 10, No. 3, September 2021, pp. 519~527
ISSN: 2252-8938, DOI: 10.11591/ijai.v10.i3.pp519-527

Boyer Moore string-match framework for a hybrid short message service spam filtering technique

Arnold Adimabua Ojugo, David Ademola Oyemade
Department of Computer Science, Federal University of Petroleum Resources Effurun, Delta State, Nigeria

Article Info

Article history:

Received Oct 1, 2020
Revised Jun 4, 2021
Accepted Jun 11, 2021

Keywords:

Boyer Moore string matching
Hybrid algorithm
Spam
Spam filters
String matching
Text processing

ABSTRACT

Advances in technology and the proliferation of mobile device have continued to advance the ubiquitous nature of computing alongside their many prowess and improved features it brings as a disruptive technology to aid information sharing amongst many online users. This popularity, usage and adoption ease, mobility, and portability of the mobile smartphone devices have allowed for its acceptability and popularity. Mobile smartphones continue to adopt the use of short messages services accompanied with a scenario for spamming to thrive. Spams are unsolicited message or inappropriate contents. An effective spam filter studies are limited as short-text message service (SMS) are 140-bytes, 160-characters, and rippled with abbreviation and slangs that further inhibits the effective training of models. The study proposes a string match algorithm used as deep learning ensemble on a hybrid spam filtering technique to normalize noisy features, expand text and use semantic dictionaries of disambiguation to train underlying learning heuristics and effectively classify SMS into legitimate and spam classes. Study uses a profile hidden Markov network to select and train the network structure and employs the deep neural network as a classifier network structure. Model achieves an accuracy of 97% with an error rate of 1.2%.

Corresponding Author:

Arnold Adimabua Ojugo
Department of Computer Science
Federal University of Petroleum Resources Effurun
P.M.B. 1221, Effurun, Warri, Delta State, Nigeria
Email: ojugo.arnold@fupre.edu.ng, maryarnoldojugo@gmail.com and arnoldojugo@gmail.com

1. INTRODUCTION

Mobile devices are redefining the communication needs, style, and infrastructure worldwide today. At its center is the mobile smartphone – with its wide acceptability due to its portability, ubiquity of services and low cost of sending messages that has consequently, promoted short text messages to become the most used means of electronic communication in the world today. Short-text message service (SMS) is the text communication service component of phone, web or mobile communication systems that uses standardized communications protocols that allow the exchange of short text messages between fixed line or mobile phone devices. The proliferation of the smartphone has today witnessed an estimated over 8 billion SMS was sent and received in Nigeria alone [1]–[3]. The tremendous rise in the usage of SMS can be attributed to the ease of use, ubiquity in nature, high open rates, low cost of transaction and inherent trust in the channel [4]–[6].

The advent of short messaging services by Neil Papworth since 1992, has seen great penetration and a tremendous growth rate of the service. Advent of mobile phones with enhance features has contributed to the large-scale adoption of SMS [3]. Increased popularity and proliferation of SMS has also seen a corresponding rise in unsolicited SMS called spams. The ITU 2005 campaign witnessed a rise in the unsolicited commercial adverts as sent to mobile phones via SMS. Recent drift from email to SMS spams is attributed to the availability of effective email filters, user awareness and industry collaboration [7], [8]. Spams are unsolicited electronic messages that include, and not limited to, emails, SMS, Voice over IP, and instant messaging from chats. Spams are unsolicited, unwanted messages from a sender and is thus, sent indiscriminately with no prior relationship to a user, mostly for commercial reasons [9]–[11]. SMS spams have since become enormous challenge – causing great loss of revenue to service providers and or mobile network operators and users in general. In total, spams have grown by over 500% from just 2015 to 2019 to billions sent and received worldwide; Implying that lots of mobile phone users are handicapped in the control of the number of spams they receive [12]. Besides being distractive and annoying, users need a certain degree of privacy with their phones and free from Spam and viruses' invasions [13]–[16]. Mobile network operators are geared towards reducing the number of spams over their network as such flooding makes the SMS channel more invasive and less secure [17].

Ojugo and Eboka [1] noted that the rising trend in the usage of SMS, which coincidentally has also caused a rise in SMS spams can be attributed to: Trust in SMS channel, high open rate, low cost of transaction, and convenient ease. SMS has great benefit for both subscribers and operators in diverse ways centered on convenience, flexibility, seamless integration of messaging services and data access. Others may include [2], [4]: delivery of notifications, guaranteed delivery, reliable, low-cost for concise data, ability to screen messages and return calls, increases productivity, more sophisticated functionality provides enhanced user benefits, delivery to multiple users at same time, ability to receive diverse information, e-mail generation, creation of user groups, integration with other data and Internet-based application, and increase in revenue for mobile network operators (MNO).

a. Sources of spams

SMS spam is generated from various sources; one of the typical spam sources is number harvesting, which is carried out by Internet sites offering "free" services. End users can also receive mobile spam from the following sources [1], [4]: organizations and individuals that pay MNO to deliver SMS to the subscribers: They are responsible for the highest number of spams received on subscriber's mobile phones. Although, MNOs have adopted and enforced use of opt-out, or even opt-in processes for the user to stop receiving promos or ads, organizations that do not pay for the SMS that are delivered to the subscribers: they are usually worse and considered as fraud because it damages MNO brands, and individual messages that disturb recipients.

Apart from the distracting and annoying effects of spam, other serious consequences generated from spams include competition for network resources otherwise allocated to hams [4]. Spams attracts extra cost to maintain effective delivery on the quality of service. Flooding the infrastructure with spams causes denial of service for hams as well as service performance degradation due to network traffic congestion [18]. MNOs are also faced with threat from malware spread via spams [19], which includes activities such as phishing and other fraud related activities prominent via social engineering [20]–[24]. These all, in turn causes financial loss, and result in damage to the mobile user's trust on the MNO's reputation [25].

b. Spam filters

Spam filters are today, saddled with real-time filtering efficiency, issues of misclassification due to high false-positives and true-negative errors, cost penalties resulting from such misclassification, and the issues of concept drift by spam makers to evade all forms of filters. Most application solutions to email spam may not be effective for SMS adoption due to [19]–[25]: conversion of email spam filters and adaptation for SMS spam implementation will lead to a downgrade in performance, SMS are limited as a 160-character of 140-bytes sized messages, and SMS are rife with a semantic structure that is rippled with slangs, symbols, emoticons, and abbreviations that will inhibit effective performance in filter training and classification support. To overcome the shortfall these challenges successfully, SMS spam filters must combine filtering techniques to reduce noise in the dataset employed for model design and training, be able to expand abbreviations and effectively translate the slangs and emoticons employed by users in SMS [26]–[29]. SMS spam filters are divided into a number of broad categories based on the method used, and these include [1]–[4]: list based, content-based, challenge/response system, collaborative and heuristics based filters. See [1], [2], [4] for more details and classification of these types. For the purpose of this study, we adopt the content-based filter – since SMS are broadcasted over a network, and they ultimately land to all valid contact mobile number identified on the network without recourse to the user for consent to deliver.

c. Hybrid semantic-content list-based (SCLB) filter

The hybrid SCLB filter combines the content-based filtering and list-based filtering technique. The content-based filtering is based on the evaluation of individual words or phrases found in message to determine if a message is a spam or not. This method analyzes message header, subject, and body to discover any distinctive characteristic [30], [31]. They are further classified into word-based and heuristic filters [1].

– Word-based filters use a set of rules to detect genuine from spam SMS. Also known as rule-filters, they use rules about actual word(s) or phrase(s) in a message to classify messages into genuine and spam classes. Rule features include word type, frequency of occurrence, structure of text (e.g., font size, color), presence of many periods between letters (e.g., F.R.E.E), and existence of image. Rules are filter-dependent and can vary from simple to very complex. A demerit of rule-based filters is that: they are knowledge intensive, time consuming process in reviewing spam messages to determine the rules and needs regular update of rules as spammers changes their tactics.

- Heuristic-based filter examines message content through various algorithms and resources and assigns points to words or phrases. Words commonly found in spams such as “FREE” or “SEX,” receive higher scores. Terms commonly found in normal messages receive lower scores. The filter then adds up total scores. If the message receives a certain score (determined by an anti-spam application administrator), the filter identifies it as spam and blocks it. Messages with score(s) lower than the target number are delivered to the use. Examples of such heuristic methods include the Bayesian filter, k-nearest neighbors classifier, AdaBoost classifier, Gary Robinson technique, support vector machine, neural network are examples. Using such heuristic filters allow many spam filtering methods to be combined in its usage; Thus, results in better performance than any single method by itself.

In addition, the list-based filter technique is described as shown in [28]–[31], [1], [4]:

- Blacklist method seeks to block unwanted messages from an already created list of senders. Blacklists are records of email addresses, internet protocol (IP) addresses and phone numbers that have been previously used to send spam. When incoming message arrives, spam filter checks if IP, email address or phone number is on a blacklist. If so, the message is considered spam and rejected. Blacklists ensure known spammers cannot reach users' inboxes. Their only demerit is that they can also misidentify legitimate senders as spammers.
- Whitelist: Rather than specify senders to block messages from, it specifies senders to allow messages from, stored in trusted-users list. Whitelist filters uses also, other techniques to cut down on the number of genuine SMS that accidentally get flagged as spam. A filter that uses just whitelist implies that anyone not approved is automatically blocked. Some anti-spams use a whitelist variation called automatic whitelist. Here, an unknown sender address is checked against a database; if they have no history of spamming – their message is delivered to the recipient's inbox and added to the whitelist.
- Gray-list: This filter works with the assumption that most spammers send batch of messages once. When message from unknown address is received, it blocks and revert a failure delivery to the sending server. If the message is resent, which most legitimate servers do, filter receives it and adds the address/phone number to the list. Although overhead of the filter is low, its demerit is the unjust delay delivery experienced by genuine messages to its recipient.

d. Statement of problem

The study is motivated by the following problems [1], [2], [32]–[37]:

- The continued rise in growth rate of spams as a means to exploit users have continued to cause both financial loss and emotional instability as consequences to users, corporate organs, and mobile network operator(s). Thus, it calls for a concerted effort to stamp out and minimize the trend in rise of spams.
- The formulation of an effective filter design has been hampered by setback(s) in SMS size limitation. These amongst other constraints have continue to create rippled impediment to feature to be selected for training and consequently contributing to poor learning and classification of learning algorithm.
- Also, SMS are rippled with slangs, abbreviations emoticons that inhibit proper classification of words.
- A major challenge in resolving spam proliferation has been, the adoption of email spam approaches to SMS filters—which has proved unsuccessful due to downgrading. There is also, the issue of heuristic adoption conflict, data encoding conflicts and filtering scheme conversion by model of choice. These have continued to hamper the performance, flexibility, and robustness of the various filtering technique.

To overcome shortfalls, we adopt Boyer Moore string-match algorithm on a hybrid semantic-content, list-based filter aimed to reduce noise in form of slangs, emoticons, abbreviations using semantic approach capable of expanding the message to enhance classification accuracy.

2. MATERIAL AND METHODS

2.1. Dataset used

We adopt the UCI machine learning repository corpus. It is a public SMS spam dataset with 5,574 instances compiled in 2012, which consists of: (a) collection of 425 SMS spam messages manually extracted from the Grumbletext web site – a UK-based forum in which cell phone users make public claims about SMS spam messages, most of them without reporting the very spam message received. The identification of the text of spam messages in the claims is a very hard and time-consuming task, and it involved carefully scanning hundreds of web pages, (b) it consists of another subset of 3,375 SMS randomly chosen ham messages of the NUS SMS Corpus (NSC), which is a dataset of about 10,000 legitimate messages collected for research at the Department of Computer Science at the National University of Singapore. The messages largely originate from Singaporeans and mostly from students attending the University. These messages were collected from volunteers who were made aware that their contributions were going to be made publicly available, and list of 450 SMS ham messages collected from Caroline Tag's PhD Thesis [38], [39]. The dataset is available at: https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection.

2.2. Boyer Moore's algorithm (BMA)

The BMA is one of the most efficient string-matching algorithms available as it actually does find matches in a sub-linear search time. It achieves this by simply scanning through the key string from left to right. On a miss, the key string is shifted a pre-computed number of characters to the right until a match of the current character occurs. Then, the next character not yet matched is considered. Since the length of the key string and the position of the current character is known, the number of characters on which the match of the key string can occur can be computed. Figure 1 matches characters depicted as upper-case letters. On inspecting, we note that the algorithm can find the exact string matches in a sub-linear search time. Its demerit(s) is that: in the search for multiple key strings in a text string - the algorithm is quite inefficient, and each key string is stored in its entirety [1].

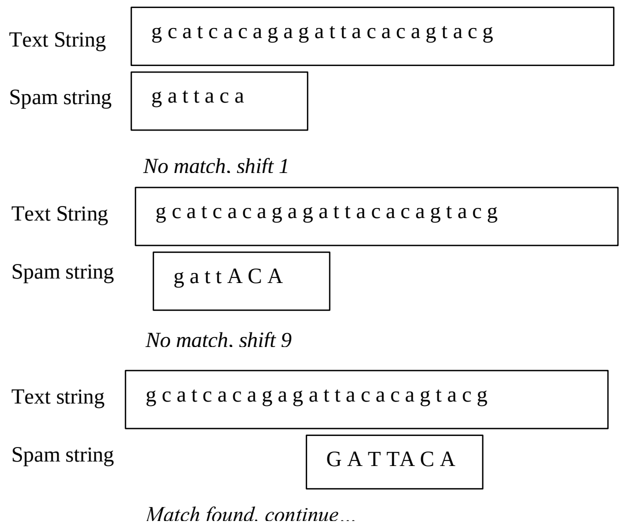

Figure 1. The Boyer Moore algorithm [1]

In traditional BMA, the model successively aligns the pattern P (spam strings) with the text T, and checks to see if the pattern string P matches the characters of T. With check complete, P is shifted right relative to T. The improved BMA has three 3 ideas not contained in its naive model as: right to left scan, bad character shift rule, and good suffix shift rule. Thus, the method examines fewer than m+n characters (at an expected sub-linear-time), and with a certain extension, runs in linear worst-case time [1], [40], as shown in Figure 1. In some scenarios, it is necessary to find a group of similar strings instead of an exact match. This is equal to a string search using wildcards. For example, instead of searching for the string perl.exe, one can search for p*rl.exe – which will also produce a result of perl.exe, parl.exe, pearl.exe. There are basically, two mechanisms that are based on two different ideas to handle character substitution or insertions namely [1]:

- Substitution error allows the replacement of one or more characters in a key search string. So, instead of searching the whole key string, only some positions are considered. This method cannot find character insertions or deletions.
- Resemblance – here, the similarities of the two strings are measured by dividing the number of characters they have in common by the length of the longer string. Formally, this equals the division of the intersection by the union. Resemblance thus, can handle character insertion and deletion (where it exists). Its drawback is that the character positions are not taken into account. So, the two strings perl.exe and lexe.rpe have resemblance one. Clearly, this is not what we want. To overcome this, the order or position numbers can be introduced.

2.3. The experimental BMA framework

Most stochastic models employs a scheme, whose procedure is thus: a) raw text are taken as input, b) it then normalized by crosschecking text against dictionaries for their root form, replace slangs and expand abbreviations using standard English words, c) it analyses normalized text to deduce their semantic concepts via language data base BabelNet repository and word sense disambiguation, d) it then breaks down the text corpus into individual element for the language processing algorithms, and e) it then defines parameters to combine result of pre-processing (original text, normalization and disambiguation stage) [1]. Conversely, BMA is quite powerful, robust, and flexible – matching strings (of texts) explained as:

- The right-to-left (RL) scan – given any set of text and a pattern string P, the traditional BMA must first seek to align the data by first checking for the occurrence of P in the text by scanning characters from right to left (rather than from left to right as in naive model). For the alignment of P against T, as shown in Figure 2 – to check if P occurs in T, BMA starts at the right end of P by: first compares T(9) with P(7). A match is found – so, it then compares T(8) with P(6), and so on and so forth, moving right to left until it finds a mismatch when comparing T(5) with P(3). At that point, the pattern string P is shifted right relative to T (the amount for the shift is discussed in the bad rule and good rule suffix). Then, the comparisons begin again at the right end of P. However, if the pattern string P is shifted one place to right after each mismatch, or after an occurrence of the pattern string P is found – then its worst-case run-time is given by O(nm) like its naive model (at which point it is unclear why we reverser engineered the model to compare characters from right to left). Cases where shifts of more than one position (large shifts) occurs, also abounds in many typical applications. These are resolved using the bad rule and good rule suffix shifts discussed below. Figure 3 shows the modified BMA framework.

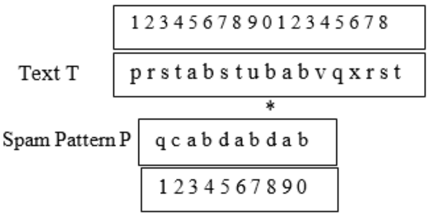

Figure 2. Modified BMA with strong good suffix shift rule

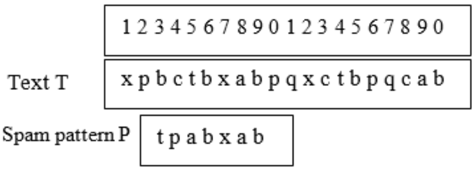

Figure 3. Modified Boyer Moore algorithm

- The bad character rule – suppose the last (rightmost) character string of the pattern P is y and the character in text T it aligns with is x ≠ y. When this initial mismatch occurs, if the rightmost position in P of character x – we can safely shift P to the right so that the rightmost x in P is below the mismatched x in T. Any shorter shift results in an immediate mismatch. Thus, the longer shift is correct (i.e., it will not shift past any occurrence of P in T). Again, if x never occurs in P, we can shift P completely past the point of mismatches in T. In these cases, some characters of T will never be examined and the method will actually run in sub-linear time. Again, if for a particular alignment of P against T, the rightmost n-i characters of P match their counterparts in T. However, the next character to the left, P(i), mismatches with its corresponding text T(k) at position k. This rule notes that P be shifted right by max[1,i-R(T(k))] places (i.e., if the rightmost occurrence in P of character T(k) is in position j < i (including the possibility of j = 0), then we shift P so that character j of P is below character k of T. Otherwise, shift P by one position. The essence of this shift rule is to shift P by more than one character when possible. As in example above, T(5) = t mismatches with P(3) and R(t) = 1. So, P is shifted right by two positions. After the shift, the comparison of P and T begins again at the right end of P [1], [2].
- Extended bad character rule: The bad character rule is useful for mismatches near the right end of P. But it has no effect if the mismatching character from T occurs in P to the right of the mismatch point. This is common when the alphabet is small and the text contains many similar, but not exact substrings. Some texts may contain different regions of high similarity. In both cases, we use extended bad character rule as it is more robust. It notes that when a mismatch occurs at position *i* of P and the mismatched character in T is x; Then, shift P to the right so that the closest x to the left of position i in P is below mismatched x in T. The extended rule gives larger shifts, and the only reason to prefer the simpler rule is its additional cost in implementing the extended rule. The simpler rule uses only O(jj) space (is the alphabet) for array R, and one table lookup for each mismatch. But the extended rule is implemented in only O(n) space, and at most one extra step per character comparison. Though, the amount of added space is not often a critical issue – the more critical question is if the longer shifts make up for the added time used by the extended rule. The original BMA only uses the simpler bad character rule [1], [2].

- The good rule suffix – The bad character rule by itself is reputed to be highly effective in practice, especially for English text; But, less effective for small alphabets as it does not lead to a linear worst-case running time. Thus, we introduce the strong good suffix rule. The original preprocessing method, for the strong good suffix rule is quite difficult and somewhat mysterious (though a weaker version, is easier to understand). In fact, the preprocessing for the strong rule was given incorrectly [1], [2]. Suppose in a given alignment of P and T, a substring t of T matches a suffix of P; But mismatch occurs at the next comparison to the left. Then find (if it exists) the rightmost copy t' of t in P such that t' is not a suffix of P and the character to the left of t' in P differs from the character to the left of t in P. Shift P to the right so that substring t' in P is below substring t' in T. If t' does not exist, then shift the left end of P past the left end of t in T by the least amount so that a prefix of the shifted pattern matches a suffix of t in T. If no such shift is possible, then shift P n-places to the right. If an occurrence of P is found, then shift P by the least amount so that a proper prefix of the shifted P matches a suffix of the occurrence of P in T. If no such shift is possible, then shift P by n-places (i.e., shift P past t in T). For example, consider the alignment of the pattern P in the text T [41], [42] in Figure 2.

When a mismatch occurs at position 8 (i.e., P(8)) and T(10) as asterisked, t = ab, and t' occurs in P starting at position P(3). Thus, P is shifted right by six places resulting in the sequence string alignment as in Figure 4 – with its listing is in Algorithm 1. Note that the extended bad character rule would have only shifted P by only one place as in the example.

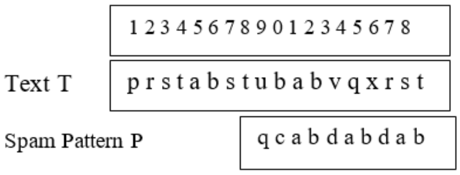

Figure 4. Modified BMA with strong good suffix shift rule

Algorithm 1. Complete pseudocode for the Boyer Moore's string pattern matching algorithm

```
Input: dataset k, cluster number cls, ham, spam, parameter s and iterations iter
Output: ham, spam
Given the pattern P, compute $L_0(i)$ and $L_1(i)$ for each position i of P
Computer R(x) for each character x 2
   // Search State
k: n
while k < m, Do
begin
i = n:
h = k:
while i > 0 && P(i) = T(h), Do
begin
i = i - 1: h = h – 1:
end
if i = 0 Then
begin
report an occurrence of P in T ending at position k
k = k + n – 1'(2)
end: else
shift P (increase k) by maximum amount determined by
invoke: (a) (extended) bad character rule && (b) good suffix rule
end:
pass pattern P in T-output to List-based Unit
classify pattern P in T as
invoke: (a) ham(class) && (b) spam(class): end:
return classification result = final output
```

The frequency probability of occurrence of each word-token(s) as counted into ham or spam – is the basis for which each incoming message is processed and classified via the string match algorithm. In the event of error in misclassification, the model goes through the text T again using the pattern P. This will automatically correct and update the database for future classification. Result of the classification of the filter into ham and spam, is the expected output of this unit.

3. FINDINGS AND DISCUSSION

3.1. Model evaluation

To measure the effectiveness and accuracy of the proposed experimental model, we measure their classification rate and improvement percentages in both training and test-dataset as summarized in Tables 1 and 2, respectively. In (1) and (2) respectively represents classification rate, and improvement percentage of true-positive instances (A) against true-negative and false-positives (B) instances. In comparison to other models, the proposed BMA hybrid SCLB filter has results displayed in Tables 1 and 2, respectively.

$$Classification\ Rate\ (CR) = \frac{No.of\ correctly\ Classified\ Rules}{No.of\ Sample\ set} \quad (1)$$

Table 1. Classification rate of each model

Model	Classification rates	
	Training data	Testing data
Naïve Bayes	23.2%	52.5%
Genetic Algorithm Trained Deep Belief Bayesian Network	48.4%	67.7%
Genetic Algorithm Trained Deep Neural Network	87.6%	91.02%
Proposed BMA Hybrid SCLB	96.89%	98.09%

$$Improvement\ Percentage = \frac{CR(A)-CR(B)}{CR(A)}\ x\ 100 \quad (2)$$

Table 2. Improvement percentage

Models	Classification improvement
Naïve Bayes	2.4%
Genetic Algorithm Trained Deep Belief Bayesian Network	5.8%
Genetic Algorithm Trained Deep Neural Network	6.9%
Proposed BMA Hybrid SCLB	7.01%

Tables 1 and 2 respectively shows classification rate with naïve bayes, GADBN and GADNN at 23.2%, 48.4%, 87.6% respectively with the proposed BMA hybrid SCLB model at 96.89%; while showing for test dataset with naïve bayes, GADBN, GADNN and proposed BMA hybrid SCLB at 52.5%, 67.7%, 91.02% and 98.09% respectively. It also shows an improvement rate classification (i.e., recovery from errors in the false-positive and true-negative classifications) respectively for naïve bayes, GADBN, GADNN and the proposed model with 2.4%, 5.8%, 6.9% and 7.01% respectively.

4. CONCLUSION

From the various approaches adopted therein by spam filters-the proposed Boyer Moore's algorithm with hybrid semantic-content, list-based filter does not require the text pre-processing, word normalization and semantic approaches as in other stochastic models. Thus, saves computational time and has also proven to be outperform others or single methods. Even with noisy data that requires text normalization and semantic expansion (multiple approaches), our hybrid yields a better output by extracting only relevant feats as parameter to train the hybrid classifier. This will inadvertently contribute to the efficiency of SMS spam filters. A great deal of concerted efforts is required to combat spams. Spam filters work by first receiving part (or all) of the message and then analyzing it in some way to decide whether it is ham (i.e., legitimate message) or spam. SMS spam filters have the capacity and granted capability to transcribe emoticons, abbreviations and slangs into standard terms as well as expand message size to enhance better feature extraction for classification algorithms and approaches. The study will also serve to reduce orthographic error found in SMS, chat groups and other social network medium that impedes learning algorithm.

REFERENCES

[1] A. A. Ojugo and A. O. Eboka, "Signature-Based Malware Detection Using Approximate Boyer Moore String Matching Algorithm," *International Journal of Mathematical Sciences and Computing*, vol. 5, no. 3, pp. 49–62, 2019, doi: 10.5815/ijmsc.2019.03.05.

[2] R. Grossi and F. Luccio, "Simple and efficient string matching with k mismatches," *Information Processing Letters*, vol. 33, pp. 113-120, 1989.

[3] J. M. G. Hidalgo, M. de B. Rodriguez, and J. C. C. Perez, “The role of word sense disambiguation in automated text categorization,” in *International Conference on Application of Natural Language to Information Systems*, Alicante, Spain, 2005, pp. 298-309, doi: 10.1007/11428817_27.

[4] A. A. Ojugo and A. O. Eboka, “Comparative evaluation for high intelligent performance adaptive model for spam phishing detection, Digital Technologies,” vol. 3, no. 1, pp. 9-15, 2018, doi: 10.1269/dt-3-1-1.

[5] C. Agwu, “The Consequences of Mobile Spam in Nigeria Emerging and Evolving Mobile Communication Sector of the Economy,” *Int. Journal of Adv. Research in Computer Science and Software Engineering*, vol. 5, no. 5, pp. 117-124, 2015.

[6] I. Murynets and R. over, “Analysis of SMS Spam in Mobility Networks,” *International Journal of Advanced Computer Science*, vol. 1, no. 1, pp. 1-8, 2013.

[7] A. A. Ojugo and D. O. Otakore, “Intelligent cluster connectionist recommender system using implicit graph friendship algorithm for social networks,” *International Journal of Artificial Intelligence*, vol. 9, no. 3, pp. 497-506, doi: 10.11591/ijai.v9.i3.pp497-506.

[8] A. A. Ojugo., R.E. Yoro., “Extending three-tier constructivist learning model for alternative delivery: ahead covid-19 pandemic in Nigeria,” *Indonesian Journal of Elect. Engineering & Computer Science*, vol. 21, no. 3, pp 1673-1682, doi: 10.11591/ijeecs.v21.i3.pp1673-1682, 2021.

[9] N. J. Yu, A. Skudrak, and Z.-L. Zhang, “Understanding SMS spam in large Cellular Network: Characteristics, Strategies and Defenses,” in *RAID 2013: Research in Attacks, Intrusions, and Defenses, International Workshop on Recent Advances in Intrusion Detection*, vol. 27, no. 6, pp. 15-26, 2013, doi: 10.1007/978-3-642-41284-4_17.

[10] T. A. Almeida, T. P. Silva, I. Santos, and J. M. G. Hidalgo, “Text normalization and semantic indexing to enhance Instant Messaging and SMS spam filtering,” *Knowledge-Based Systems*, vol. 108, no. 15, pp. 25-32, 2016, doi: 10.1016/j.knosys.2016.05.001.

[11] D. Oyemade and A. A. Ojugo, “A property oriented pandemic surviving trading model,” *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 5, pp. 7397-7404, 2020, doi: 10.30534/ijatcse/2020/71952020.

[12] H. Sajedi, G. Z. Parast, and F. Akbari, “SMS Spam Filtering Using Machine Learning Techniques: A Survey,” *Machine Learning Research*, vol. 1, no. 1, pp. 1-14, 2016, doi: 10.11648/j.mlr.20160101.11.

[13] A. Narayan and P. Saxena, “The curse of 140 characters: Evaluating the efficacy of SMS spam detection on Android,” in *Proceedings of the Third ACM workshop on Security and privacy in smartphones & mobile devices*, Berlin, Germany, 2014, pp.33–42, doi: 10.1145/2516760.2516772.

[14] M. T. Nuruzzaman, C. Lee, M. F. A. bin Abdullah, and D. Choi, “Simple SMS spam filtering on independent mobile phone,” *Journal of Security and Communication Networks*, vol. 5, no. 10, pp. 1209-1220, 2012, doi: 10.1002/sec.577.

[15] A. A. Ojugo and A. O. Eboka, “Mitigating technical challenges via redesigning campus network for greater efficiency, scalability and robustness: a logical view,” *International Journal of Modern Education & Computer Science*, vol. 6, pp. 29-45, 2020, doi: 10.5815/ijmecs.2020.06.03.

[16] D. Cook, J. Hartnett, K. R. Manderson, and J. Scanlan, “Catching Spam before it arrives: Domain Specific Dynamic Blacklists,” in *Proceedings of the Fourth Australasian Symposium on Grid Computing and e-Research (AusGrid 2006) and the Fourth Australasian Information Security Workshop (Network Security) (AISW 2006)*, Hobart, Tasmania, Australia, January 2006, vol. 54, pp. 193-202, doi: 10.1145/1151828.1151851.

[17] B. Han and T. Baldwin, “Lexical Normalisation of Short Text Messages,” in *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, USA 2011, pp. 368-378.

[18] O. B. Longe, A. Robert, S. C. Chiemeke, and F. O. Ojo, “Features outliers and their effects on the efficiencies of text classifiers in the domain of electronic mail,” *The Journal of Computer Science and its Applications*, vol. 15, no. 2, pp. 1-8, 2008.

[19] I. A. Gheyas and L. S. Smith, “Feature subset selection in large dimensionality domains,” *Pattern Recognition*, vol. 43, no. 1, pp. 5-13, 2010, doi: 10.1016/j.patcog.2009.06.009.

[20] S. Hsib, M. Motwani, and A. Saxena, “Anti-Spam methodologies: a comparative Study,” *International Journal of Computer Science and Information Technologies*, vol. 3, no. 6, pp. 5341-5345, 2012.

[21] S.-S. Hong, W. Lee, and M.-M. Han, “The Feature Selection Method based on Genetic Algorithm for Efficient of Text Clustering and Text Classification,” *International Journal of Advances in Soft Computing and its Applications*, vol. 7, no. 1, pp. 23-40, 2015.

[22] A. A. Ojugo and A. Eboka, “Inventory management and prediction using market basket analysis associative rule mining: memetic algorithm approach,” *International Journal of Informatics and Communication Technology (IJ-ICT)*, vol. 8, no. 3, pp. 128-138, doi: 10.11591/ijict.v8i3.pp128-138.

[23] C. Kobus, F. Yvon, and G. Damnati, “Normalizing SMS: are two metaphors better than one?,” In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, Morristown, NJ, USA, 2008, pp 441–448.

[24] C. Wang, *et al*., “A behavior-based SMS antispam system,” *IBM Journal of Research and Development*, vol. 54, no. 6, pp. 3:1-3:16, Nov.-Dec. 2010, doi: 10.1147/JRD.2010.2066050.

[25] P. G. Juneja and R. K. Pateriya, “A Survey on Email Spam Types and Spam Filtering Techniques,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 3, no. 3, pp. 2309-2314, 2014.

[26] T. B. Shahi and A. Yadav, “Mobile SMS Spam Filtering for Nepali Text using Naïve Bayesian and Support Vector Machine,” *International Journal of Intelligence Science*, vol. 4, no. 1, pp. 24-28, 2014, doi: 10.4236/ijis.2013.41004.

[27] S. Delany, C. Padraig, T. Alexey, and C. Lorcan, "A case-based technique for tracking concept drift in spam filtering," *Knowledge-Based Systems*, 32, pp. 187-195, 2004.

[28] N. Desai and M. Narvekar, "Normalization of Noisy Text Data," *Procedia Computer Science*, vol. 45, pp. 127-132, 2015, doi: 10.1016/j.procs.2015.03.104.

[29] T. Dayarante, T. Chaminda, H. K. N. Amarasingha, and J. M. R. S. Jayakody, "Content-based hybrid SMS spam filtering system," in *Proceedings of ITRU Research Symposium*, University of Moratuwa, Sri Lanka, 2013, pp. 31-35.

[30] G. Sethi and V. Bhootna, "SMS Spam Filtering Application using Android," *International Journal for Computer Science and Information Technologies*, vol. 5, no. 3, pp. 1424-1426, 2014.

[31] G. V. Cormack, J. M. G. Hidalgo, and E. P. Sanz, "Spam Filtering for Short Messages," in *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007*, Lisbon, Portugal, 2007, pp. 313-320, doi: 10.1145/1321440.1321486.

[32] Q. Xu, E. W. Xiang, Q. Yang, J. Du, and J. Zhong, "SMS Spam Detection Using Noncontent Features," in *IEEE Intelligent Systems*, vol. 27, no. 6, pp. 44-51, Nov.-Dec. 2012, doi: 10.1109/MIS.2012.3.

[33] A. K. Uysal, S. Gunal, S. Ergin, E. Sora Gunal, "The impact of feature extraction and selection on SMS spam filtering," *Journal Elektronika IR Elektrotechnika*, vol. 19, no. 5, pp. 67–72, 2013.

[34] R. Daoud and I. Jebril, "Computer virus strategies and detection methods," *International Journal of Open Problems Computational Mathematics*, vol. 1, no. 2, pp. 46-56, 2008

[35] W. Cuker, S. Cody, and E. Nesselroth, "Genres of spam: expectations," *Progress in Intelligence Computing and Applications*, vol. 2, no. 1, pp. 22-33, 2013, doi: 10.4156/pica.vol2.issue1.2.

[36] G. Wittel and F. Wu, "On attacking statistical spam filters," *In Proceedings of First Conference on Email and Anti-Spam*, CSEA ' 2004, 44, pp. 37-45, 2004.

[37] A. A. Ojugo and R. E. Yoro, "Forging a machine learning intrusion detection framework to curb the distributed denial of service attack," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 2, pp. 126-132, 2021, doi: 10.11591/ijece.v11i2.pp1498-1509.

[38] I. Androulidakis, V. Vlachos, and A. Papanikolaou, "FIMESS: filtering mobile external SMS spam," in *6th Balcan Conference in Informatics (BCI 2013)*, Thessaloniki, Greece, 2013, pp. 221-227, doi: 10.1145/2490257.2490288.

[39] J. M. G. Hidalgo, G. C. Bringas, E. P. Sanz, and F. C. Garcia, "Content-based SMS Spam filtering," in *Proceedings of the 2006 ACM Symposium on Document Engineering*, Amsterdam, The Netherlands, 2006, pp. 107-114, doi: 10.1145/1166160.1166191.

[40] A. A. Ojugo and O. D. Otakore, "Forging an optimized Bayesian network model with selected parameter for detection of the Coronavirus in Delta State of Nigeria," *Journal of Applied Science, Engineering, Technology and Education*, vol. 3, no. 1, pp. 37-45, 2020, doi: 10.35877/454RI.asci2163.

[41] A. A. Ojugo and A. Eboka, "Memetic algorithm for short messaging service spam filter text normalization and semantic approach," *International Journal of Informatics and Communication Technology (IJ-ICT)*, vol. 9, no. 1, pp. 13-27, 2020, doi: 10.11591/ijict.v9i1.pp9-18.

[42] A. A. Ojugo, A. O. Eboka, R. E. Yoro, M. O. Yerokun, and F. N. Efozia, "Hybrid Model for Early Diabetes Diagnosis," *2015 Second International Conference on Mathematics and Computers in Sciences and in Industry (MCSI)*, Sliema, Malta, 2015, pp. 55-65, doi: 10.1109/MCSI.2015.35.