An Improved Multi-Class Classification Algorithm based on Association Classification Approach and its Application to Spam Emails

Rami Mustafa A. Mohammad

***Abstract* – In our everyday life, we may use email for personal and professional matters. Yet, email benefits have been bedeviled with the remarkable use of annoying, harmful, and fraudulent messages that commonly referred to as spam emails. Several anti-spam campaigns hover around machine learning and data mining techniques were devised in literature. An intelligent data mining approach referred to as Associative Classification (AC) presents itself as a possible method that might efficiently identify spam emails. In this study, an improved Spam Classification based on the Association Classification algorithm (SCAC) is proposed. In addition to the robust rule generation procedure, the improved model creation process, and the enhanced prediction mechanism, the SCAC algorithm is able to derive a new class value that doesn't exist in the original dataset that is the "Uncertain" class value. Hence, the SCAC algorithm doesn't introduce several contributions in the field of AC only but also it has contributed to the spam detection domain. These contributions have mutually reinforced the superb classification abilities of the SCAC algorithm when compared to several other intelligent techniques.**

***Index Terms*— Association, Classification, Spam, e-mail, Multi-Class**

I. INTRODUCTION AND MOTIVATION

A recent statistical study [1], revealed that in 2018 the number of email users is anticipated to reach over 3.8 billion users. This represents an increase of 100 million users compared to the previous year. The study stated that in 2009 the number of email users is estimated at 1.9 billion users. This indicates that the amount of email users has doubled in less than 10 years. Actually, the estimated population of the world in 2018 is 7.4 billion persons. That represents more than half of the whole world uses email. In 2022, the number is projected to reach more than 4.2 billion. In fact, the number of email accounts is more than the number of users in the sense that some users may have more than one account ("on average 1.75 accounts"). Business emails represent more than 25% of all email accounts. More than 80% of professionals picked emails as their preferred communication channel. More than 150 billion emails are sent every day.

Manuscript received August 12, 2019; revised January 1, 2020. Rami Mustafa A Mohammad is now with the Department of Computer Information Systems, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O.Box 1982, Dammam, Saudi Arabia. (e-mail: rmmohammad@iau.edu.sa)

These statistics shed light on how important the email is in people's social and professional endeavors. Nevertheless, the email benefits have been ruined with the exceptional use of irritating, destructive, dishonest and deceptive emails that are commonly dispatched indiscriminately by a person or by a party that has no direct relationship with the recipient. Such emails are normally denoted as spam emails. One might wonder, why would someone open an email coming from an unknown sender. In fact, the "From" field and the "subject" field in an email are the main promoters to decide whether to open an email or permanently delete it. Users tend to open an email if the "from" field shows that the email is originated from a well-known sender [2]. However, sometimes users lean towards opening and reading their inbox emails even if these emails were received from unknown sender due to the human curiosity nature. Most importantly, it has been shown that 33% of users open an email if it has a fascinating subject matter regardless of whether the sender is known or not [3]. The people who send spam emails are commonly known as "Spammers". Such people are smart enough to exploit any chance for originating attractive spam emails. Spammers normally combine social engineering and tactical tricks when sending spam emails [4] [5]. Let us recall how spammers could take advantage of "Hurricane Irma" for spreading their spam emails in 2017 [6]. Nepal earthquake is another example that can be cited in this regard [7]. These disasters have had calamitous consequences not only on the people where this disaster happened but also on the worldwide human community. Yet, these disasters were like a gift for spammers. Sporting events are another favorite subject of spammers. FIFA World Cup 2018 is an example of such events [8]. The insolence of spammers has reached the extent of harassing famous people without paying attention to the social, financial, or political positions of those honest people. Of such, the spam emails that used the name of the First Lady of the US "Melania Trump" [9]. Promotional emails are another way to lure victims. Actually, it has been emphasized that these kinds of emails are the most commonly used among others followed by adult-related advertising. Spam emails related to financial matters ranked in third place [10]. In general, spammers are smart people who exploit any possible opportunity to deceive innocent people. So, they are expected to take advantage of any news of interest to the public to spread spam emails. It should be mentioned that a recent study showed that most of the spam emails are originated from the U.S., China, and Russia [11].

Spam emails are normally written in English. However, in 2018 the ratio of English written spam emails decreased to 90% after reaching 96% a few years ago [12]. The decrease of English written spam emails is offset by the increase of spam emails written in other languages. Hence, one can conclude that spam emails are getting more "international". The risks posed by spam emails involve several other threats such as scam, fraud, and phishing websites [13] [4] [14]. Belarus generates the most spam per capita [12]. Spam emails accounted for 45% of email traffic worldwide [10]. This means that almost half of the emails that are transmitted daily are spam emails. This ratio looks terrifying, especially if we know that the number of transmitted emails per day in 2017 is almost 269 billion emails, more than 121 billion emails of them were spam email. This means that the worldwide per capita of spam emails reaches up to 16 spam emails every day. The good news is that spammers receive 1 reply for every 12 million spam emails. Yet, with so few people replying, one may ask what stimulates launching spam email campaigns? Surprisingly, it has been found that over the course of one-year spammers may earn more than 3 million US dollars even with only one reply per 12 million spam emails sent [10]. In 2012, spam emails cost business organizations more than 20 billion dollars. This amount is expected to surpass 250 billion dollars.

Several spam countermeasures have been devised in literature. Filtering spam emails is an essential step towards protecting users from being spammed. Section II in this article is dedicated to reviewing the latest researches on combating spam emails. In addition to many different techniques, this article focuses on intelligent approaches based primarily on Machine Learning (ML) and Data Mining (DM) methods. DM and ML approaches proved their merits in several security domains [15] [16] [17] [18] [19] [20] [21]. Spam is considered a supervised classification problem. Classification is a commonly employed task in the DM that has a learning method frequently denoted to as supervised learning paradigm because it aims to create a model (classifier) from a training dataset that comprises a set of examples each of which has a set of input features along with a corresponding class value. In addition to others, Association Classification (AC) is one of the algorithms that fall under the umbrella of the supervised classification DM approach. In general, AC offers two unique aspects over different commonly used classification techniques. The first could be its capabilities to produce relatively easy rules that may be easily understood and also can be manually modified by the model creator. The second thing is that this technique typically discovers extra valuable concealed details skipped by other classification methods, hence, the classification accuracy in the generated classifier can be increased. One possible reason behind delivering the extra valuable details is due to the fact that AC uses association rule finding strategies during the training stage [22].

In this article, an improved *"Spam Classification based on Association Classification algorithm (SCAC)"* is proposed. Besides the inherited advantages of using AC techniques, the SCAC algorithm is expected to generate a classifier that includes robust and effective rules that would help in classifying spam emails. Such rules should be selected carefully from the set of generated rules during the learning process. In fact, one motivation behind creating the SCAC algorithm is to enhance the rule generation and the model creation processes. Yet, as soon as the rules are selected and the model is created they should be utilized in an optimal way in order to obtain the best classification results. This, in fact, another motivation behind creating the SCAC algorithm in the sense that the SCAC algorithm will also enhance the prediction mechanism. Nonetheless, in addition to several contributions that the SCAC will make in the field of AC *(particularly in the rule generation procedure, the model creation process, and the prediction mechanisms)* SCAC will also make a remarkable contribution in the field of spam classification. Specifically, the SCAC algorithm is capable to derive a new class value that doesn't exist in the original dataset that is the *"Uncertain"* class value. This indeed will increase the number of correctly classified examples which will inevitably escalate the expected accuracy ratio. In general, any improvement in the classification of spam emails is considered an achievement, even if it is a marginal improvement, especially if we took into account the huge number of spam emails distributed over the internet on a daily base. Section V of this article is dedicated to assessing the overall performance of the SCAC algorithm. The performance of SCAC will be compared against several DM algorithms using different classification metrics that are commonly used when creating intelligent spam detection models.

The remainder of the article is organized as follows: Section II revises relevant research paper and shed light on diverse methods utilized for identifying spam emails. In Section III, the suggested algorithm is explained in detail. Section IV is dedicated on assessing the performance of the suggested technique. Section V and Section VI discusses the results obtained in the experimental section. Lastly, the overall conclusions will be discussed in Section VII.

II. RELATED WORK

Filtering spam email messages before they reach the users' inboxes is the ultimate goal of all anti-spam methods including the intelligent approaches that are mainly based on DM and ML techniques.

Typically, the first line of defense against spam emails is the email user himself. Some techniques depend on the honest and dedicated work of email users in reporting unsolicited emails [23] [24]. Such methods are normally called *"The Community Reporting Approach"*. The fundamental principle of this approach is *"Honestly, evaluate and report any spam email you have encountered in your inbox so that other users can be warned"*. Hence, this approach relays on the users' experience in identifying spam emails. However, this throws a great burden on the email users to decide whether a specific email is a sincere or a hostile one. The users are also evaluated according to their history in accurately reporting spam emails. The better the user's reputation is, the more reliable his reports will be. Yet, this method is a tedious process and a user should devote a considerable amount of his time in studying and

labeling their inbox emails. Most importantly, spammers know that the weakest chain in any protection system is the human factor [25] [26]. In addition, users may check their inbox to search for spam emails and if they find any, they would delete it instead of reporting it. Hence, some important information may be lost, while in fact, this information can be useful for service providers when deciding if a specific email is a spam and whether or not to automatically place it in the spam folder. Generally, the more a user reports spam emails, the better spam filter a service provider can deliver. Most importantly, the users still did not do the full tasks required from them in the sense that they normally report *"False Negative"* (FN) emails *("spam emails that are wrongly classified as valid")*, but few of them may report *"False Positive"* (FP) emails *("genuine emails that are wrongly classified as spam")*. Therefore, in order to do their job properly, users have to check not only the inbox but also all other folders such as junk and spam folders.

Another solution to combating spam campaigns is the enactment of legal measures. In 2001, Argentina was the first country to enact and implement laws to criminalize spam emails [27]. Many countries have also enacted laws that criminalize spammers. Yet, legal measures are difficult to apply in practice because spammers can launch their spam campaigns and then disappear into cyberspace. Nowadays, several for-profit, non-profit, and research studies are offered with the aim of combating spam emails with minimal user intervention. The *"Spamhaus project"* is an example of non-profit organizations that track spam and other related threats such as malware, botnet, and most importantly phishing [28]. However, doing a quick search on the internet, one can find lots of commercial anti-spam tools. In general, the success of anti-spam tools depends on recognizing spam emails correctly before reaching the user's inbox. In addition to other content, emails are mostly made up of text. So, it makes sense to use text mining classification techniques to combat spam emails. Several DM and ML methods have been used for developing text mining-based spam detection models such as Naïve Bayes [29] [30], Neural Network [31], Support Vector Machine [32], and Nearest Neighbor [33]. In general, text mining-based spam detection methods start by collecting two datasets of labeled emails, i.e. a dataset of spam emails and a dataset of genuine emails. Such emails are decomposed into separate tokens or words. Then the appearance ratio of each word is calculated in each dataset. Once a new unseen email is acquired, the set of tokens (words) which usually appear more frequently are thought as being signs that the email is spam or not.

As a rule of thumb, misclassifying spam emails as genuine (False Negative) is more expensive than incorrectly classifying genuine emails as spams (False Positive). Therefore, several intelligent methods have been devised with the aim of minimizing the false-negative rate. Spam emails classification is commonly considered as a binary classification problem because each email can be either spam or honest. Several ML algorithms have proven to be applicable in binary classification domains such as spam emails detection. Of such, Logistic Regression [34], Neural Network [35], Support Vector Machine [36]. Naïve Bayes has also been successfully utilized in filtering spam emails. One of the prominent research studies that utilized Naïve Bayes is the one that attempted to combine three Naïve Bayes classifiers to facilitate enhancing the performance of traditional Naïve Bayes in classifying spam emails [37]. The first classifier separates the training dataset into two classes, i.e. *"spam"* and *"non-spam"*. In the second classifier, linear programming is utilized for optimizing the decision thresholds that define the fine line that separates spam from non-spam emails. However, the third classifier combines the results of the first and the second classifiers. The experimental results showed that this technique surpasses the traditional Naïve Bayes algorithm. Naïve Bayes has also produced good results when incorporated with feature extraction algorithms such as Cost-Sensitive Multi-objective Genetic Programming [38]. Neural Networks have actively contributed in detecting spam emails. The work done by Ozgur et al [39], proved this claim. Yet the results were not encouraging. The capabilities of the interval "type-2 fuzzy sets" in predicting spam emails have been assessed [40]. This system gave the users the ability to pick which type of spams he wishes to block by simply enabling the dictionary related to that type.

A server-side plugin for filtering spam emails has been created and is called SpamGuru [41]. SpamGuru gives a score that ranges from 0 to 1000 for each email. The higher the score, the more dangerous the email is. Users can also report spam emails so that the plugins can improve their performance by learning continuously. The SpamGuru creates a folder that contains a set of emails that the plugin couldn't give a definite decision whether they are spam or not. As soon as an email is marked as a spam email, SpamGuru provides four possible options:

1- Permanently delete the confirmed spam email.
2- Archive it.
3- Direct it to challenge queue which provides a challenge/response authentication of the sender ID.
4- Tag it as possible spam and delivered it to the user so that he can decide the appropriate action.

As mentioned earlier, the current research study assumes that spam is a supervised classification problem.

However, some research studies treated spam emails as an unsupervised classification problem where a set of unlabeled examples are used for creating the classification model. An example of the unsupervised classification models is the work done in [42] which essentially assumes that spam emails normally belong to a *"spam campaign"* and they are rarely sent individually. Hence, spam emails can be identified using *"campaign signature"*. This study defines spam campaigns as a set of highly related emails that are reported by many users. Hence, the spam campaign is normally started by creating a single spam email and then make too many copies of it by keeping some parts intact and obfuscating others with the aim of luring the email users. Recognizing the intact and obfuscated parts is an essential step towards identifying the campaign signature. The experimental evaluation showed competitive results when compared to some other supervised classification algorithms.

A research study [43] conducted in 2007 aimed to compare the performance of four different DM and ML algorithms those are *"Decision Trees", "Naïve Bayesian", "Support Vector Machine", and "Neural Network".* Different evaluation criteria have been used for the assessment process. The experimental evaluation showed that the Decision Tree and Naïve Bayesian classifiers produced better results than Neural Network and Support Vector Machine.

Some research studies investigated the applicability of DM and ML techniques in predicting the non-English Language written spam emails. For instance, the work done in 2009 [44] aimed to assess the performance of six different DM and ML algorithms in filtering spam emails written in Arabic. The study concluded that some extra features should be added so as to improve the accuracy of predicting spam emails written in Arabic. Another study has investigated the Turkish Language written spam emails [45]. The researchers employed text contents and raw contents in an email to decide whether an email is a genuine one or not. for experimental purposes, the researchers have utilized the Adabost ensemble method and the obtained results were very promising.

In 2012, an Adaptive Neural Fuzzy Interference System (ANFIS) has been proposed for classifying spam emails [46] where five criteria are used for identifying spam emails those are: number of common timestamps, presence of URL or Hyperlinks, text priority, number of times marked as spam, and number of associated user pages. Fuzzy Rule Induction Algorithm (FURIA) has been proposed in 2006 for classifying spam emails [47]. This algorithm is an improvement to the well-known RIPPER algorithm. The experimental results revealed that FURIA surpassed RIPPER and Decision Tree in most cases. An improved Bayesian algorithm for filtering spam emails was proposed in 2009 [48]. In this study, boosting algorithms were utilized for making the Bayesian algorithm stronger.

III. THE PROPOSED SPAM CLASSIFICATION BASED ASSOCIATION CLASSIFICATION ALGORITHM

In this section, the proposed Spam Classification based Association Classification algorithm is explained thoroughly including the rules generation phase, the final model (classifier) creation phase, and the prediction phase. The first phase i.e. rules generation phase consists of two sub-phases namely: 1- discovering frequent rule-items 2- rules generation. In the way of accomplishing this phase, the proposed algorithm will look for any generated rules having the same rule-antecedent but are associated with different class values (rule consequent) to create the so-called multi-class rules. The final model creation phase starts by sorting the generated rules, selects the most effective rules, and remove the set of unnecessary (redundant) rules. After this phase, a model (classifier) that can work in both single-class domains and multi-class domains is created. The last phase, i.e. prediction phase, sheds light on how to predict the class value of the of unseen testing instances so as to evaluate the overall capability of the suggested technique.

The algorithm assumes that the input features hold categorical or discrete numerical (non-continuous) values. Yet, as a preliminary step, users can utilize any discretization method if any input feature holds continuous values. Discretization is the way of placing values within categorize in order to decrease the volume of available states an attribute contains. The generated categorize are processed as discrete value(s). Additional information regarding discretization could be obtained from [49]. Normally, when dealing with attributes of different units and scales an applicable normalization technique must be utilized to reduce the chances of outweighing attributes that hold greater ranges and to ensure that the model is going to converge to better outcomes. In particular, time and date related features might hold a greater range of data in case they are presented as numerical form. Amongst the commonly employed tactics is to scale dataset value(s) in a pre-determined range(s). For example [0...1] or [-1...1]. A well-known method that can be used is the *"max-min"* normalization technique [50]. The SCAC algorithm is illustrated in Fig. 1, and is detailed in Sub-Sections A, B, and C.

Input: *TDS "Training Dataset", MS "Minimum Support", MC "Minimum Confidence"*

Output: A Classification Model

1. Rules Generation Phase
- *a. Find frequent rule-items*
- *b. Develop a new rule for each rule-item passes a predefined MS. Then evaluate these rules and ignore anyone that doesn't pass the predefined MC*

2. Final Model Creation Phase
- *a. Sorting the developed rules*
- *b. Remove unwanted rules*

3. Prediction Phase
- *a. Predict the class value of testing examples*

Fig. 1. The Proposed Algorithm Pseudo Code

Two terms should be defined before explaining the different phases of the SCAC algorithm those are:

1- Support: the frequency of input value(s) that is/are linked with particular output value from the size of the dataset.

2- Confidence: the frequency of input value(s) that is/are linked with particular output value from the frequency of the input value(s).

A. Rules Generation Phase

This phase starts by finding frequent rule-items. Any input feature passes a predefined support threshold (Minimum Support (MS)) is thought a possible frequent rule-item. Normally, association classification methods determine frequent rule-items either horizontally or vertically [51]. Most association classification algorithms employ a horizontal technique for finding frequent rule-items [22]. The horizontal method finds frequent rule-items iteratively. In other words, this technique goes over the training dataset several times and the frequent rule-items determined at iteration *i* are used to specifying the frequent rule-items at iteration *i+1*. For instance, the frequent rule-items determined in the first

iteration are used for determining the frequent rule-items in the second iteration, and the rule-items determined in the second iteration are used for determining the frequent rule-items in the third iteration and so forth.

On the other hand, the vertical method creates an array that contains the location of each input feature value (possible frequent rule-item) in the training dataset. Such an array is usually called Transaction ID (TID) and is used for calculating the support for each input feature by finding the length (how many times a specific feature occurs in the training dataset) of the TID. The vertical technique is more effective than horizontal technique in finding frequent rule-items because vertical technique follows an efficient method for calculating the support for each possible frequent rule-item [51] [22]. The SCAC algorithm utilizes a vertical technique in finding frequent rule-items by making simple intersections between TIDs. Several algorithms have already employed a vertical technique for identifying frequent rule-items such as MCAR [52] and CACA [53]. Yet, the proposed algorithm in this research differs from others by considering the value of the class variable in the sense that it intersects only TIDs that are sharing the same class value. Hence, the proposed algorithm minimizes the number of intersections which would result in speeding-up the rules creation phase and the algorithm learning in general. Firstly, the algorithm finds the frequent rule-items of size 1 (one feature value). Next, it intersects the TIDs of the disjoint rule-items of size1 to find the possible rule-items of size 2 (two feature values). Then the algorithm finds the rule-items of size 3 (three feature values) by intersecting the TIDs of the disjoint rule-items of size 2, and so forth. To elaborate further, let consider the training dataset shown in Table I. Such a dataset contains 2 input features *F1* and *F2* as illustrated in Table I.

TABLE I
A BINARY TRAINING DATASET

No	F1	F2	Class
1	x2	y1	c1
2	x1	y2	c1
3	x2	y1	c2
4	x2	y2	c2
5	x1	y1	c1
6	x1	y2	c1
7	x3	y1	c2
8	x2	y2	c1
9	x2	y1	c2
10	x1	y2	c1
11	x1	y1	c2
12	x2	y1	c1

Let assume that the predefined minimum support value (MS) is 20%. The algorithm starts by finding TIDs of the frequent rule-items of size 1 as shown in Table II and Table III. These tables show that the candidate rule-items that are linked with class c1 are x1, x2, y1, and y2 because their support values pass the predefined MS. However, x3 is also linked with class c1 but it is discarded because its support value is less than the predefined MS.

TABLE II
SIZE 1 POSSIBLE RULE-ITEMS FOR F1

Possible Rule-item	TID	Length	Support
$<x_1>, c_1$	**2,5,6,10**	**4**	**33.33%**
$<x_1>, c_2$	11	1	8.33%
$<x_2>, c_1$	**1,8,12**	**3**	**25.00%**
$<x_2>, c_2$	**3,4,9**	**3**	**25.00%**
$<x_3>, c_1$	-	0	0.00%
$<x_3>, c_2$	7	1	8.33%

TABLE III
SIZE 1 POSSIBLE RULE-ITEMS TID FOR F2

Possible Rule-item	TID	Length	Support
$<y_1>, c_1$	**1,5,12**	**3**	**25.00%**
$<y_1>, c_2$	**3,7,9,11**	**4**	**33.33%**
$<y_2>, c_1$	**2,6,8,10**	**4**	**33.33%**
$<y_2>, c_2$	4	1	8.33%

Therefore, the algorithm produces a new rule for each selected rule-item (size 1 rules) as revealed in Table IV.

TABLE IV
GENERATED RULES OF SIZE 1 FOR CLASSIFYING C1

no	Selected rule-item	Generated rule
1	x1	if <x1> ⟶ c1
2	x2	if <x2> ⟶ c1
3	y1	if <y1> ⟶ c1
4	y2	if <y2> ⟶ c1

On the other hand, Table II and Table III show that x_1, x_2, x_3, y_1, and y_2 are linked with class value c_2. Yet, the selected rule-items are x_2 and y_1 only because their support value is greater than the predefined MS. Hence, new rules are generated as shown in Table V.

TABLE V
RULES OF SIZE 1 FOR CLASSIFYING C2

no	Selected rule-item	Generated rule
1	x2	if <x2> ⟶ c2
2	y1	if <y1> ⟶ c2

From Table IV and Table V, one can comprehend that x_1 and y_2 are linked with one class value only. Yet, x_2 and y_1 are linked with multiple class values. Going back to Table II and Table III, we can see that x_2 and y_1 have 2 support values that pass the predefined MS each of which is linked with a different class value. Typically, traditional association classification algorithms maintain the rule-item that is mostly linked with the class variable and ignore others even if they have a support value greater than the MS. However, the proposed algorithm acts differently since it considers all cases that pass the predefined MS. Such a situation results in producing multi-class rules. This situation is one of the advantages the SCAC algorithm might deliver. Several classification domains might benefit from this. In other words, the SCAC algorithm can be applied to several classification domains. The classification domain considered in this study, i.e. spam classification, is an example of such domains.

As soon as all possible size 1 rules are produced, the SCAC will intersect the TIDs of the selected rule-items with the aim of identifying the next possible rule-items and generate the size 2 rules. As mentioned earlier, the algorithm intersects only TIDs of rule-items that are having the same class value. This

indeed would save time by ignoring unnecessary intersection processes. Table VI shows the results of the intersection processes.

TABLE VI
POTENTIAL SIZE 2 RULE-ITEMS

Possible Rule-item	TID	Length	Support
$<x_1, y_1>, c_1$	5	1	8.33%
$\mathbf{<x_1, y_2>, c_1}$	**2, 6, 10**	**3**	**25.00%**
$<x_2, y_1>, c_1$	1, 12	2	16.67%
$<x_2, y_2>, c_1$	8	1	8.33%
$<x_2, y_1>, c_2$	3, 9	2	16.67%

Results in Table VI reveal that only one intersection has passed the predefined MS, i.e. the intersection that is linked to class c_1 and involves x_1 and y_2. Hence, the algorithm generates the following rule:

if $<x_1$ && $y_2> \longrightarrow c_1$

As soon as the algorithm generates all possible rules, it will assess each rule and the ones that pass predefined minimum confidence (MC) are maintained and all others are ignored.

It is well known that the association classification algorithms produce rules in the *"if..then"* form statements. Normally, *"if"* statements execute until the first *"TRUE"* or until covering all possible statements. In other words, in the implementation phase, the *"if"* statement that comes first has a higher priority. The main concern is the mechanism the algorithm will be using in devising the final decision (identifying the class value) with the presence of some rules connected with multiple classes. For instance, let us go back to Table IV and Table V above, and let assume that the generated rules are implemented in the same order they are shown in these tables starting from Table IV. Now, consider an instance that holds x_3 in the first input feature and y_1 in the second input feature and the actual class value is c_2. Then the algorithm will produce a wrong classification result because it will use rule number three in Table IV. However, if the algorithm could reach rule number 2 in Table III then it will produce a correct answer, but the algorithm was not able to reach this rule because it stops evaluating the generated rules once it finds the first *"TRUE"*. The SCAC algorithm solves this problem by merging the rules that are having the same antecedent and linked with different classes. Not only this but also it examines the training dataset and finds the frequency of each class associated with the rule antecedent and places the one that has a higher frequency at the beginning of the rule consequent. For instance, in the example introduced above the algorithm generates the following rule:

if $<y_2> \longrightarrow c_2 \mid c_1$

In this rule, c_2 is placed before c_1 in the rule consequent because it is linked 4 times with the rule antecedent, i.e. y_1 in the training dataset, whereas c_1 is linked 3 times only. Nevertheless, if the algorithm finds a case where all classes have the same frequencies it places them randomly in the rule consequent.

However, for binary classification domains such as phishing [54], Autism [55], or even the classification problem that is discussed in this research, i.e. spam classification if a rule produces two values that means all possible class values can be assigned for the testing instance. Here the uniqueness of the SCAC algorithm is manifested in the sense that the SCAC algorithm would produce one extra class value that is not defined in the original dataset. Such class value is called "*Uncertain*" and is translated in different meanings depending on the classification problem insight. For instance, in the case of phishing, the "*Uncertain*" is translated as *"Suspicious website"*. Yet, in the case of spam classification, it is translated as *"a conservative decision on the email status"*.

B. Model (Classifier) Creation Phase

Normally, Association Classification methods generate a great number of rules [22]. Some of such are significant and effectively contribute to producing a model that can be generalized. However, some rules, on the other hand, are considered unnecessary (redundant) and the generalization ability of the model is not adversely affected if such rules are deleted. This phase aims to create the final model and it includes two sub-phases namely: rule sorting and rule pruning. Once all possible rule(s) are devised, the algorithm will sort them based on some criteria with the aim of choosing and adding the most relevant ones to the model and discarding others. Hence, ensure that the produced model has enough power to generalize. There are three commonly used criteria for sorting the generated rules, namely: *"rule confidence"*, *"rule support"* and *"rule length"* [22]. Rule confidence and rule support can be easily computed for single class rules. However, for multi-class rules, the process is a bit confusing. For example, consider the following rule *(which generated in the previous example)*:

if $<y_2> \longrightarrow c_2 \mid c_1$

The question is, which class should be considered when calculating the rule confidence and rule support, is it c_1 or c_2? The SCAC algorithm answers this question by calculating the rule confidence and rule support for every single class and then it averages the results. This way ensures fairness when sorting the rules.

Firstly, the algorithm sorts the rules using the first criterion, i.e. *"rule confidence"*. If more than one rule has the same confidence, the algorithm will sort the rules according to the next criterion i.e. rule support. Further, if more than one rule has the same rule confidence and rule support, the rules are sorted as per the rule length. However, in more severe cases where several rules have the similar rule confidence, rule support, and rule length, the SCAC algorithm suggests adding one more criterion that is the class-correlation criterion. This criterion calculates how many times (frequency) the rule is linked with the class value. The rule that has a higher class-correlation is favored over the others. In the case of multi-class rules, the algorithm finds the class-correlation for each class and then it finds the average. In the worst case, if more than one rule having the same rule confidence, rule support, rule

length, and class-correlation, then the algorithm sorts the rules randomly. As soon as all generated rules are sorted, the second sub-phase (pruning phase) starts. The SCAC algorithm uses the training dataset for evaluating the sorted rules one by one starting with the first sorted rule. If the rule covers at least a single training instance it will be appended to the final model, otherwise, it will be discarded. All examples that are covered by the rule will be removed and the remaining examples will be used for evaluating the next rule. The process is reiterated until covering all training instances or all rules are assessed. If all training examples are covered but there are some rules that are not evaluated, they will be removed (pruned). However, if all rules are assessed but some training examples are not covered, the algorithm defines a new rule called default-class. The default class assigns the same class value for all remaining training examples which is the class with the higher frequency in the remaining set of examples not in the training dataset. In general, this sub-phase aims to select the minimal set of rules that can contribute to producing a model with high generalization abilities.

C. Prediction Phase

The training dataset is used for fulfilling the first two phases, i.e. rule generation phase, and model creation phase. However, in the third phase, the testing dataset is used for assessing the performance of the generated model.

Normally, classification data mining algorithms employ one rule for predicting the class value of a specific testing dataset example [22] [5]. However, the SCAC algorithm employs multiple rules for forecasting the class label. The main advantage of allowing several rules to contribute in forecasting the class value is that if one rule made an error, the others may correct this error. The algorithm inspects the testing dataset instances one by one. For each testing instance (unseen example) the algorithm collects the group of rules which completely match the antecedent part of the testing example regardless of the class value (consequent part) associated with the rule. Several rules may be collected, each of which may be linked with a different class value. Then, the algorithm groups the set of rules that are linked with the same class value. Hence, several groups may be formed where each group is linked with a specific class value. Nevertheless, if only one group is created, this means that there is a consensus on the class value of the training example. On the other hand, if more than one group is created, the algorithm finds the harmonic mean of the confidence value for every group. The group that gets the highest harmonic mean is assumed to be linked with the correct class value. Therefore, such class value is assigned to the testing dataset example. The rationale behind using harmonic mean in this research is because it is not affected much by the fluctuation of the confidence value of each rule in the group [56]. In addition, it gives a chance for the rules with small confidence values to participate in devising the final class. To find the harmonic mean for a specific group the algorithm firstly adds the reciprocals of the confidence *(f)* of each rule, divide the sum by *(r)*, then take the reciprocal of the result as sown in equation 1:

$$\textit{Harmonic Mean} = \frac{r}{\frac{1}{f1}+\frac{1}{f2}+\frac{1}{f3}+\cdots+\frac{1}{fr}} \qquad (1)$$

If two or more groups produced the highest harmonic mean, the algorithm favors the group that has the maximum number of rules (group length). In a more severe case, if two or more groups have the same harmonic mean and the same number of rules then the SCAC algorithm produces one extra class value that is not defined in the original dataset. This class value is called *"Uncertain"*. In this case, the final decision on the email status is left for the users. One extra benefit of the proposed class value assignment method is that it reduces the utilization of the default rule. Hence, improves the performance of the SCAC algorithm by minimizing the error rate. Recall that the default rule is created for a set of training instances that are not covered by any single rule.

IV. EMPIRICAL EVALUATION

The process of distinguishing spam emails from non-spam ones is a typical classification problem in the sense that several attributes are utilized to acquire hidden knowledge. Such knowledge is considered the classification model that is used for identifying spam emails.

Typically, there are two possible values where an email can be classified into, namely: Spam or Non-Spam. The SCAC algorithm is intelligently capable to generate rules associated with more than one class value, i.e. spam and non-spam. In other words, the SCAC algorithm can produce a new label that doesn't exist in the original dataset. Such class value is called *"Uncertain"*. If an email is classified as *"Uncertain"*, that means it is either:

1- A non-spam email that is distorted by a set of suspicious attributes.
2- A spam email that is cleverly decorated with a set of attributes that give a sense of legitimacy on it.

In either case, there must be a third party that takes the initiative of making the final decision on the status of the email, which is normally the user himself.

This section aims to evaluate if the SCAC algorithm is capable to create good classifier(s). All experiments were performed in the *"Waikato Environment for Knowledge Analysis"* (WEKA) software [57]. WEKA is an open-source Java tool and is commonly used for evaluating DM and ML models. WEKA includes several DM and ML algorithms. In addition, it can be used for doing a lot of data preprocessing tasks including feature selection, normalization, discretization, etc. The set of experiments were executed on a system with *"CPU Pentium Intel® Core™ i5-2430M @ 2.40 GHz, RAM 4.00 GB"*. The environment was the *"Windows 10 64-bit operating system"*.

A. Describing the Training Dataset

The dataset employed in this study is the commonly utilized spam dataset [58]. This dataset contains 57 attributes and 1 class attribute that represents the status of every dataset example i.e. "spam" or "non-spam", denoted by 1 or 0, respectively. The dataset comprises 4601 instances, 2788 of

which are non-spam emails and 1813 are spam emails. Additional details about the dataset might be found in [58].

B. Algorithms Used for Comparison and Experimental Settings

Several Association Classification (AC) and Rule Mining based Classification algorithms will be utilized to assess the applicability and the performance of the SCAC algorithm on the well-known and commonly utilized benchmark spam dataset that is acquired from UCI repository [58]. Besides SCAC, the set of algorithms utilized in the experiments are CMAR [59], MCAR [52], MCAR2 [60], CBA [61], decision tree (C4.5) [62], PART [63], RIPPER [64], eDRI [65], and MAC [66]. The SCAC algorithm has been implemented in Java. The CMAR algorithm is obtained from the AC community [67]. On the other hand, the MCAR, MCAR2, eDRI, and MAC were obtained from their prospective authors. However, the Weka built-in versions of CBA, C4.5, PART, and RIPPER are used in the experiments. The CMAR, MCAR, MCAR2, eDRI, and MAC were selected because they represent the state-of-the-art multi-class association classification algorithms. Yet, the rationale behind selecting the CBA, C4.5, PART, and RIPPER is since they are all rule-based algorithms. In addition, they devote different approaches in generating classifiers, and they are frequently applied in building classification models [4] [49]. In general, all the comparable algorithms share an essential feature with the SCAC in the sense that they produce rule-based models.

Determining the MS and MC threshold values is an essential step towards carrying out the experiments. Several scholars in the field have stressed that the MS threshold normally controls the number of rule(s) generated and the time spent through the rule creation phase and the model (classifier) construction phase. However, there is no agreement on specific MS and MC threshold values that can be employed in all cases. Therefore, following several research studies such as [68] [52] [59] [61] [22] [69] the MS threshold value is set to 2%. However, the MC threshold is set to 50%.

C. Validation Technique

The ten-fold validation technique is used for evaluating the classification models produced by SCAC and all other considered algorithms. This technique starts by dividing the training dataset arbitrary into 10 parts where 9 parts are employed for learning the model and the remaining one i.e. the 10^{th} part is used for testing the produced model and is normally referred to as *("hold-out dataset")*. The process is repeated 10 times on the training dataset and the achieved results are then averaged. Random shuffling is performed with the aim of ensuring that all classes are available in each part of the training datasets and the hold-out testing dataset. Such a process is normally called *"Stratification"*. In other words, the stratification process guarantees that all class values exist when splitting the original dataset.

D. Evaluating Criteria

There are four classification possibilities when evaluating any supervised classification model namely: True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). General definitions of these classification possibilities can be found in [4] [25] [70] [49]. Yet, in previous works related to intelligent spam classification approaches, these classification probabilities have specific definitions. For instance, TP denotes the number of non-spam emails accurately labeled as non-spam. TN signifies the number of spam emails accurately labeled as spam. FP represents the number of non-spam emails incorrectly labeled as spam. FN represents the number of spam emails incorrectly labeled as non-spam. However, the proposed algorithm in this research will handle the spam classification problem as a multi-class classification issue not as a binary classification issue because it might produce one extra class that is the "*Uncertain*" label.

In other words, in the current research, the spam problem will be treated as a binary classification problem by the contrasted DM algorithms and as a multi-class classification problem by the SCAC algorithm. Hence, there should be unified criteria for evaluating the overall performance of the SCAC algorithm in addition to all other DM algorithms considered in this research. Such evaluation criteria should be consistent with previous works related to binary and multi-class classification problems in general and to classifying spam emails using DM and ML techniques in particular. Classification accuracy (ACU) is commonly used classification criterion in binary and multi-class classification domains [59] [54] [70] and is calculated as per equation 2. ACU is described as the percentage of accurately classified instances compared with the total amount of instances in the testing dataset *(Knowing that the examples that are classified as "Uncertain" are also considered correctly classified examples)*.

$$ACU = \frac{Number\ of\ Correctly\ Classified\ Eaxamples}{Total\ Number\ of\ Examples} \dots (2)$$

The number of the generated rules is also an important criterion for evaluating the overall performance of the produced models for both binary classification domain and multi-class classification domains. This criterion gives a better reading about the complexity of the produced models.

The next section will discuss the results obtained from the SCAC algorithm and all other considered DM algorithms.

V. RESULTS ANALYSIS

Table VII summarizes the classification accuracy (%) yielded by the SCAC algorithm and other considered algorithms for the spam dataset.

The results depicted in Table VII clearly show that the SCAC algorithm surpasses all other AC and rule mining-based classification algorithms in forecasting the class value of the examples in the testing dataset.

In particular, the SCAC algorithm achieved an accuracy rate that outperformed CMAR, MCAR, MCAR2, CBA, C4.5, RIPPER, PART, eDRI, and MAC with margins of 2.06%, 0.73%, 1.49%, 1.49%, 2.08%, 0.88%, 2.28%, and 0.47% respectively. The third column in Table VII shows the number of correctly classified cases. This column shows that SCAC classified 94, 33,94, 68, 68, 95, 40, 104, and 21 more examples

than CMAR, MCAR, MCAR2, CBA, C4.5, RIPPER, PART, eDRI, and MAC respectively.

TABLE VII
ACCURACIES OBTAINED FROM SCAC AND OTHER CONSIDERED ALGORITHMS

Algorithm	Accuracy %	Correctly Classified Examples
CMAR	92.41%	4252
MCAR	93.74%	4313
MCAR2	92.41%	4252
CBA	92.98%	4278
C4.5	92.98%	4278
RIPPER	92.39%	4251
PART	93.59%	4306
eDRI	92.19%	4242
MAC	94.00%	4325
SCAC	94.47%	4346

In general, the classification accuracies obtained from SCAC and all other considered classification algorithms are acceptable and that reflects the suitability of rule-based algorithms in classifying spam emails. In other words, rule-based algorithms have exceptional capabilities in exploring training data. It should be stated here that out of 4346 correctly classified examples there were 37 examples classified as *("uncertain")*. In fact, this result revealed that deriving a new class value is not the only player in getting such an outstanding classification result.

For instance, the SCAC algorithm correctly classified only 33 examples more than several multi-class classification algorithms those are CMAR, MCAR, MCAR2, eDRI, and MAC which means that these algorithms were able to correctly classify some examples that the SCAC algorithm has classified as *("uncertain")*. Recall that the SCAC algorithm improves the classification accuracy by reducing the utilization of the default rule i.e. assign a default value for the set of examples that are not covered by any single rule during the prediction phase. Actually, the CMAR, MCAR, MCAR2, eDRI, and MAC algorithms make what is assumed a conservative decision by assigning *"spam"* class value for any example that is not covered by any generated rule in the sense that they assume that the FP decision is less expensive than the TP one.

In other words, the consequences of assigning a *"spam"* label for non-spam emails have far fewer negative effects than assigning a *"non-spam*" label for a spam email. Nonetheless, unlike the SCAC algorithm, these algorithms utilize a guessing technique rather than a scientific method in determining the class label of these examples. To conclude, the CMAR, MCAR, MCAR2, eDRI, and MAC algorithm were lucky in assigning a label that is matching the label of the instances that have not been covered by any rule during the prediction phase.

Overall, although it was one of the main reasons as to why the SCAC algorithm achieved comparable results, deriving a new class value that never exists in the training dataset is not the only reason behind achieving such results but there were several other players crucially contributed in attaining such a remarkable classification performance those are:

- The robust rule generation procedure
- The improved model creation process
- And, the enhanced prediction mechanisms

As for the rule generation procedure, the SCAC algorithm is capable not just to assign a single class label for each rule but further it can find all possible labels and offer them a disjunctive manner. Typically, traditional AC algorithms snub such extra rules while in fact besides satisfying the end-users needs, they may positively contribute to improving the classification abilities of the produced model. In other words, the SCAC algorithm has exceptional abilities in generating some rules that are capable to accurately forecast the label of the test examples, particularly the set of examples that are suspicious, while the other DM techniques have misclassified them just because those test examples were assigned one class value, i.e. spam or non-spam. In the model creation process, the SCAC algorithm sorts the produced rules in a way that ensures selecting the most relevant rules and ignores what assumed redundant ones according to some sorting criteria. Actually, the SCAC algorithm recommends a new sorting criterion that is the *("class-correlation")*. Such a criterion plays an important role in producing a model with good generalization abilities and that was confirmed from the obtained classification performance of the SCAC algorithm. The prediction mechanism is another reason for the high accuracy achieved by the SCAC algorithm because it was capable to produce correct predictions in the sense that multiple rules are employed for predicting the class value of the testing examples. In other words, all rules applicable to a testing dataset example are considered in producing the final decision on the email status. Hence, if one rule made an error the others may correct that error. An interesting result that one can comprehend from Table VII is that RIPPER produced the worst results.

That can be attributed due to the fact that the RIPPER algorithm discovers the rules greedily and prunes rules by means of gradual decreased error pruning technique [64] [49] which results in removing some rules that may be useful in forecasting the email class. The results depicted in Table VII showed that several multi-class association classification algorithms, i.e. MCAR, MAC, and SCAC produced better results than the traditional (single class) association classification algorithm, i.e. CBA. That can be attributed because these algorithms have the ability to drill down the training dataset in a way that allows for devising robust rules and as a result more robust classification model. Fig. 2, shows the number of rules produced by the SCAC and other considered DM algorithms.

The figure stresses that the CMAR, MACR, eDRI, MAC, and SCAC algorithms produced a large number of rules if compared to MCAR2, C4.5, RIPPER, or PART. In fact, this has been inherited from AC techniques that use the training example several times in learning the rules, not like other traditional classification techniques that use the training example only once for a specific rule. That clarifies the smaller models created by other considered classification algorithms.

However, the MCAR2 produced the minimum number of rules among other multi-class classification algorithms. This can be attributed because the MCAR2 utilized five advanced rule pruning methods that remove some rules that might still

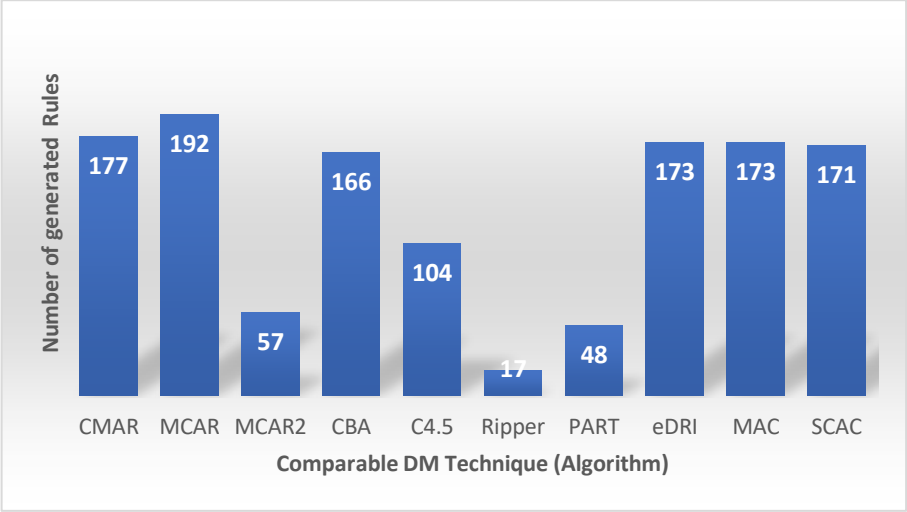

Fig. 2. Number of Rules obtained from SCAC and other considered algorithms

useful in producing more accurate models and that was reflected on the prediction accuracy of MCAR2. Overall, the SCAC algorithm was capable to generate multi-label rules from the spam dataset. Such rules are connected to a new class that doesn't previously defined in the training dataset. These rules have contributed positively in raising the prediction accuracy rate because they were able to manage testing instances that are neither spam nor non-spam (suspicious emails). The fact that the SCAC algorithm extracts new rules is a good sign on the capability of the algorithm in exploring dataset insight in a way that most recent AC methods are not able to do.

VI. EFFICIENCY ANALYSIS OF SCAC ALGORITHM

The rules generation phase plays a vital role in defining the efficiency of the proposed algorithm. Therefore, this section compares the rules generation phase for both the SCAC algorithm and the well-known traditional AC algorithm that is the CBA. Particularly, this section computes how many times the disjoint rule-items were merged in each iteration in SCAC and CBA. The obtained results depicted in Table VIII show that in the first iteration both algorithms have done the same number of rule-items merging. However, in the next iterations (i.e. from the second iteration until the seventh iteration) the results showed significant improvements. For instance, the total number of rule-items merging in the SCAC algorithm is decreased by 26.9% when compared to CBA. These results prove that the rules generation phase in the SCAC algorithm has improved upon the traditional AC approach.

TABLE VIII
NUMBER TIMES THE DISJOINT RULE-ITEMS WERE MERGED IN EACH ITERATION

	SCAC	CBA
Iteration #1	55	55
Iteration #2	105	278
Iteration #3	655	1345
Iteration #4	1430	3825
Iteration #5	2099	6892
Iteration #6	2100	8135
Iteration #7	0	3451
Total	6444	23981

Overall, the SCAC algorithm has offered a new technique of avoiding a lot of redundant merging processes which may cause a reduction in the execution time which is very crucial to judge on the overall efficiency of any software-based solutions. Consequently, the memory usage might also decrease. Digging deeply, it has been shown that the time need for building the final model using the SCAC algorithm has considerably reduced from 48624ms in CBA to 7851ms with a margin of 83.85%. This improvement is mainly due to the fact the SCAC does not do unnecessary merging for rule-items that are associated with different class labels. When comparing the memory usage of the SCAC algorithm against that needed for the CBA algorithm it was obvious that the SCAC algorithm is more efficient in the sense that memory usage has decreased by almost 45%. Overall, the experimental outcomes showed that the SCAC algorithm is more efficient than the conventional AC algorithm. This is mainly because of the innovative rule generation process employed by the SCAC algorithm.

VII. CONCLUSIONS AND FUTURE WORKS

Spam emails is an important issue facing the email users' community due to its clear impact on both financial and personal levels. Association classification is a promising data mining technique that enticed scholars because of the high predictive abilities and the easy to understand models it develops. The spam email classification problem was considered in this research and a novel AC algorithm has been developed and named SCAC algorithm. Unlike traditional AC algorithms, the SCAC algorithm learns new rule(s) that are associated with more than a class value. As shown in the experimental section, the SCAC algorithm produced comparable results when contrasted against several other classification algorithms using various evaluation measures. Most importantly, SCAC was capable to extract multi-label rules from the spam dataset. Such rules were linked with a new class label, i.e. "*Uncertain"* which did not originally appear in the training dataset. These rules enhanced further the SCAC classification ability. In general, deriving new class value from the training dataset is not the only contribution that the SCAC

algorithm introduces. It is indeed the main contribution that the article presents but also the SCAC algorithm includes several other aspects. For instance, the SCAC algorithm contributed in the rule generation phase because it considers the value of the class value when intersecting the TIDs. In other words, it only intersects the TIDs that are sharing the same class value. Hence, the proposed algorithm minimizes the number of intersections. In addition, the SCAC algorithm contributed to the model (classifier) creation phase which consists of two sub-phases namely: rule sorting and rule pruning. The contributions that the SCAC algorithm made on these two sub-phases are discussed in detail in section B. Not only this but also, the SCAC algorithm contributed to the prediction phase as described in Section C. The high prediction results are also a contribution in the sense that the SCAC algorithm achieved comparable results when compared to other DM and ML algorithms. It is worth stating that the SCAC algorithm is an offline classification algorithm. In other words, the classification model created by the SCAC algorithm cannot be changed once it is created. Nonetheless, in order to facilitate lifelong learning, a possible future work is to improve the SCAC algorithm so that it can learn new knowledge and update the created model as soon as a new training dataset becomes available. Several classification domains may benefit from this such as phishing websites. Applying the SCAC algorithm to other domains is also one of the possible future works with the aim of getting a clearer picture of its abilities.

REFERENCES

[1] H. Tschabitscher, "Worldwide Email Statistics," Lifewire, 23 March 2018. [Online]. Available: https://www.lifewire.com/how-many-email-users-are-there-1171213. [Accessed 1 May 2018].

[2] R. M. Mohammad, F. Thabtah and L. McCluskey, "Tutorial and critical analysis of phishing websites methods," Computer Science Review-ELSEVIER, vol. 17, no. 1, pp. 1-24, 2015-A.

[3] H. Tschabitscher, "Fascinating email facts," Lifewire, 23 March 2018. [Online]. Available: https://www.lifewire.com/how-many-emails-are-sent-every-day-1171210. [Accessed 1 May 2018 A].

[4] R. M. Mohammad, F. Thabtah and L. McCluskey, "Predicting phishing websites based on self-structuring neural network," Neural Computing and Applications, vol. 25, no. 2, pp. 443-458, 2013-B.

[5] R. M. Mohammad, F. Thabtah and L. McCluskey, "Intelligent Rule based Phishing Websites Classification," IET Information Security, vol. 8, no. 3, pp. 153-160, July 2013-A.

[6] N. Alfred, "Watch out for Hurricane Harvey phishing scams," CBS NEWS, 30 8 2017. [Online]. Available: https://www.cbsnews.com/news/hurricane-harvey-phishing-scams-cybercriminals/. [Accessed 2 5 2018].

[7] F. Allego, "Threat Encyclopedia," TREND MICRO, 5 5 2015. [Online]. Available: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/spam/608/spammers-use-earthquake-in-nepal-for-scam-donation-funds. [Accessed 2 5 2018].

[8] J. Bettencourt , "Kaspersky Lab Spam and Phishing report: FIFA 2018 and Bitcoin among 2017's most luring topics," Kaspersky, 15 2 2018. [Online]. Available: https://usa.kaspersky.com/about/press-releases/2018_fifa-2018-and-bitcoin-among-2017-most-luring-topics. [Accessed 6 5 2018].

[9] A. IOVINE, "Email scam claims that Melania Trump wants to give you 20 million dollars," AOL, 29 3 2017. [Online]. Available: https://www.aol.com/article/news/2017/03/29/email-scam-claims-that-melania-trump-wants-to-give-you-20-millio/22017185/. [Accessed 6 5 2018].

[10] E. Bauer, "15 Outrageous Email Spam Statistics that Still Ring True in 2018," Propeller , 1 2 2018. [Online]. Available: https://www.propellercrm.com/blog/email-spam-statistics. [Accessed 6 5 2018].

[11] G. Darya, V. Maria and S. Tatyana, "Spam and phishing in 2017," SecureList, 15 2 2018. [Online]. Available: https://securelist.com/spam-and-phishing-in-2017/83833/. [Accessed 6 5 2018].

[12] S. Statistics, "Spam Statistics," Spam Statistics, 2018. [Online]. Available: https://antispamengine.com/spam-statistics/. [Accessed 6 5 2018].

[13] R. M. Mohammad, F. Thabtah and L. McCluskey, "An Improved Self-Structuring Neural Network," in Pacific Asia Knowledge Discovery and Data Mining Conference (PAKDD) 2016, Auckland, 2016-B.

[14] R. M. Mohammad, F. Thabtah and L. McCluskey, "An assessment of features related to phishing websites using an automated technique," in International Conference for Internet Technology and Secured Transactions, 2012, London, 2012.

[15] M. Nijim and H. Albataineh, "En-Stor: Energy-Aware Hybrid Mobile Storage System using Predictive Prefetching and Data Mining Engine," Engineering Letters, vol. 26, no. 2, pp. 252-256, 2018.

[16] Sukono, J. Saputra, B. Subartini, J. Hotmaida, S. Supian and Y. Hidayat, "An Application of Genetic Algorithm Approach and Cobb-Douglas Model for Predicting the Gross Regional Domestic Product by Expenditure-Based in Indonesia," Engineering Letters, vol. 27, no. 3, pp. 411-420, 2019.

[17] K. M. Alallayah, W. AbdElwahed, M. Amin and A. H. Alhamami, "Attack of Against Simplified Data Encryption Standard Cipher System Using Neural Networks," Journal of Computer Science, vol. 6, no. 1, pp. 29-35, 2010.

[18] R. M. Mohammad, "A Neural Network based Digital Forensics Classification," in 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA), Aqaba, 2018.

[19] R. M. Mohammad and M. Alqahtani, "A comparison of machine learning techniques for file system forensics analysis," Journal of Information Security and Applications, vol. 46, no. 1, pp. 53-61, 2019.

[20] R. Mohammad, "Investigating Trust Issue in Semantic Web Applications," in The Second forum in Information Security-Naif Arab University for Security Sciences, Riyadh-Saudi Arabia, 2016.

[21] R. M. Mohammad, "An Enhanced Multiclass Support Vector Machine Model and its Application to Classifying File Systems Affected by a Digital Crime," Journal of King Saud University - Computer and Information Sciences, 2019.

[22] F. Thabtah, "Review on Associative Classification Mining," The Knowledge Engineering Review, vol. 22, no. 1, pp. 37-65, 2007.

[23] V. V. Prakash and A. O'Donnell, "Fighting Spam with Reputation Systems," Queue - Social Computing, p. 50, 9 Nov 2005.

[24] E. Zheleva, A. Kolcz and L. Getoor, "Trusting spam reporters: A reporter-based reputation system for email filtering," ACM Transactions on Information Systems (TOIS), vol. 27, no. 1, 2008.

[25] R. Mohammad, "An Ensemble Self-Structuring Neural Network Approach to Solving Classification Problems with Virtual Concept Drift and its Application to Phishing Websites," University of Huddersfield, 2016-A.

[26] R. M. Mohammad, F. Thabtah and L. McCluskey, "Predicting Phishing Websites using Neural Network trained with Back-Propagation," in ICAI, Las Vigas, 2013-C.

[27] M. D'Auro and I. de Achaval, "Data protection in Argentina: overview," 2014. [Online]. Available: http://www.ebv.com.ar/images/publicaciones/trdatap.pdf. [Accessed 5 11 2018].

[28] Spamhaus, "Spamhaus," Spamhaus , 1998. [Online]. Available: https://www.spamhaus.org/. [Accessed 12 5 2018].

[29] P. Graham, "Better Bayesian Filtering," in Spam Conference, 2003.

[30] G.-C. J, "People and spam," in The Spam Conference, 2005.

[31] Q. Ma, Z. Qin, F. Zhang and Q. Liu, "Text spam neural network classification algorithm," in International Conference on Communications, Circuits and Systems (ICCCAS), 2010 , Chengdu, China, 2010.

[32] R. Shams and R. M. E, "Classifying Spam Emails Using Text and Readability Features," in IEEE 13th International Conference on Data Mining (ICDM), 2013 , Dallas, TX, USA, 2013.

[33] R. M. Aliguliyev, R. M. Aliguliyev and S. Nazirova, "Classification of Textual E-Mail Spam Using Data Mining Techniques," Applied Computational Intelligence and Soft Computing, p. 8, 2011.

[34] G. V. Cormack and M. D. Smucker, "Efficient and effective spam filtering and re-ranking for large web datasets," Information Retrieval, vol. 14, no. 5, p. 441–465, 2011.

[35] S. Singh, A. Chand and S. L. Pranit, "Improving Spam Detection Using Neural Networks Trained by Memetic Algorithm," in Fifth International

Conference on Computational Intelligence, Modelling and Simulation (CIMSim), 2013, Seoul, South Korea, 2013.
[36] S. O. Olatunji, "Improved email spam detection model based on support vector machines," Neural Computing and Application, pp. 1-9, 2017.
[37] Y. Song, A. Kolcz and C. Lee Giles, "Better naive bayes classification for high-precision spam detection," Journal of Software: Practice and Experience (SPE), vol. 39, no. 11, pp. 1003-1024, 2009.
[38] Z. Yang, L. HongYu, M. Niranjan and P. Rockett, "Applying Cost-Sensitive Multiobjective Genetic Programming to Feature Extraction for Spam E-mail Filtering," in European Conference on Genetic Programming, 2008.
[39] L. Özgür, T. Güngör and F. Gürgen, "Spam Mail Detection Using Artificial Neural Network and Bayesian Filter," in International Conference on Intelligent Data Engineering and Automated Learning, 2004.
[40] R. Ariaeinejad and A. Sadeghian, "Spam detection system: A new approach based on interval type-2 fuzzy sets," in Canadian Conference on Electrical and Computer Engineering (CCECE), 2011 24th, Niagara Falls, ON, Canada.
[41] R. Segal, J. Crawford, J. Kephart and B. Leiba, "SpamGuru: An enterprise anti-spam filtering system," in The First Conference on E-mail and Anti-Spam (CEAS 2004), Mountain View, CA, 2004.
[42] F. Qian, A. Pathak, Y. Charlie Hu, Z. Morley Mao and Y. Xie, "A case for unsupervised-learning-based spam filtering," in The ACM SIGMETRICS international conference on Measurement and modeling of computer systems, NEW YORK, 2010.
[43] S. Youn and D. McLeod, "A Comparative Study for Email Classification," in Advances and Innovations in Systems, Computing Sciences and Software Engineering, Dordrecht, 2007.
[44] A. El-Halees, "Filtering Spam E-Mail from Mixed Arabic and English Messages: A Comparison of Machine Learning Techniques," International Arab Journal of Information Technology (IAJIT), vol. 6, no. 1, pp. 52-59, 2009.
[45] A. Çıltık and T. Güngör, "Time-efficient spam e-mail filtering using n-gram models," Pattern Recognition Letters, vol. 29, no. 1, pp. 19-33, 2007.
[46] D. Kalbande, H. Panchal, N. Swaminathan and P. Ramaraj, "ANFIS based Spam filtering model for Social Networking Websites," International Journal of Computer Applications, vol. 44, no. 11, p. 0975 – 8887, 2012.
[47] J. Hühn and E. Hüllermeier, "FURIA: an algorithm for unordered fuzzy rule induction," Data Mining and Knowledge Discovery, vol. 19, no. 3, p. 293–319, 2009.
[48] P.-y. Liu, L.-w. Zhang and Z.-f. Zhu, "Research on E-mail Filtering Based On Improved Bayesian," JOURNAL OF COMPUTERS, vol. 4, no. 3, pp. 271-275, 2009.
[49] I. H. Witten, E. Frank and A. H. Mark, Data mining: practical machine learning tools and techniques with Java implementations, Third ed., Morgan Kaufmann, 2011.
[50] J. Anil, N. Karthik and R. Arun, "Score normalization in multimodal biometric systems," Pattern Recognition, pp. 2270-2285, 2005.
[51] M. J. Zaki and K. Gouda, "Fast vertical mining using diffsets," in The ninth ACM SIGKDD international conference on Knowledge discovery and data mining, Washington, D.C., 2003.
[52] F. Thabtah, C. Peter and Y. Peng, "MCAR: Multi-class Classification based on Association Rule," in The 3rd ACS/IEEE International Conference on Computer Systems and Applications., 2005.
[53] Z. Tang and Q. Liao, "A New Class Based Associative Classification Algorithm," IAENG International Journal of Applied Mathematics, vol. 36, no. 2, pp. 15-19, 2007.
[54] R. M. Mohammad and H. Y. AbuMansour, "An intelligent model for trustworthiness evaluation in semantic web applications," in 8th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 2017.
[55] F. Thabtah, "Autism Spectrum Disorder Screening: Machine Learning Adaptation and DSM-5 Fulfillment," in ICMHI '17 Proceedings of the 1st International Conference on Medical and Health Informatics 2017, Taichung City, Taiwan, 2017.
[56] B. Jeuris and R. Vandebril, "Geometric Mean Algorithms Based on Harmonic and Arithmetic Iterations," in International Conference on Geometric Science of Information, 2013.
[57] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, "Waikato Environment for Knowledge Analysis," University of Waikato, 2011. [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/. [Accessed 20 December 2011].
[58] M. Hopkins, E. Reeber, G. Forman and J. Suermondt, "Spambase Data Set," Hewlett-Packard Labs, 1 7 1999. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/spambase.
[59] W. Li , J. Han and J. Pei, "CMAR: accurate and efficient classification based on multiple class-association rules," in IEEE International Conference on Data Mining , San Jose, CA, USA, 2001.
[60] H. Abu Mansour, Rule pruning and prediction methods for associative classification approach in data mining, Huddersfield: University of Huddersfield, 2012.
[61] B. Liu, H. Wynne and M. Yiming, "Integrating classification and association rule mining," in KDD'98 Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, New York, NY, 1998.
[62] J. R. Quinlan and M. Kaufmann, "C4.5: Programs for Machine Learning," Machine Learning, vol. 16, no. 3, pp. 235-240, 1993.
[63] E. Frank and I. H. Witten, "Generating Accurate Rule Sets Without Global Optimization," in ICML '98 Proceedings of the Fifteenth International Conference on Machine Learning, 1998.
[64] W. W. Cohen, "Fast Effective Rule Induction," in Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, 1995.
[65] I. Qabajeh, F. Chiclana and F. Thabtah, "Constrained dynamic rule induction learning," Expert Systems with Applications, pp. 74-85, 2016.
[66] N. Abdelhamid, A. Ayesh, S. Ahmadi, W. Hadi and F. Thabtah, "MAC: A Multiclass Associative Classification Algorithm," Journal of Information & Knowledge Management, pp. 1-10, 2013.
[67] W. Packages, "WEKA Packages," WEKA, 2011. [Online]. Available: weka.sourceforge.net/packageMetaData/. [Accessed 2 4 2013].
[68] G. Costa, R. Ortale and E. Ritacco, "X-Class: Associative Classification of XML Documents by Structure," ACM Transactions on Information Systems, p. 40, 2013.
[69] B. Liu, Y. Ma and C. K. Wong, "Improving an Association Rule Based Classifier," in Principles of Data Mining and Knowledge Discovery. PKDD 2000, 2000.
[70] H. Y. Mansour and H. A. Alshihri, "Adapting associative classification for detecting phishing websites," in The First Summit on Countering Cyber Crimes, Riyadh, 2015.

Rami Mustafa A Mohammad received a B.S. from the Department of Computer Science and Information Systems, Philadelphia University in 1998, and an M. S. from the Arab Academy for Banking and Financial sciences in 2002. In 2016, he received his Ph.D. from the Department of Computing and Engineering, University of Huddersfield (UK). He is now an assistant professor in the Department of Computer Information Systems, Imam Abdulrahman Bin Faisal University. His research interests include Phishing Detection, Spam Detection, Data Analysis and Mining, and Digital Forensics.