Digital Technologies, 2018, Vol. 3, No. 1, 9-15
Available online at http://pubs.sciepub.com/dt/3/1/2

DOI:10.12691/dt-3-1-2

Comparative Evaluation for High Intelligent Performance Adaptive Model for Spam Phishing Detection

A.A. Ojugo[1,*], A.O. Eboka[2]

[1]Department of Mathematics/Computer Science, Federal University of Petroleum Resources Effurun, Delta State, Nigeria
[2]Department of Computer Science Education, Federal College of Education Technical, Asaba, Delta State, Nigeria
*Corresponding author: ojugo.arnold@fupre.edu.ng, ojugoarnold@hotmail.co.uk, arnoldojugo@yahoo.com

Received September 19, 2018; Revised November 02, 2018; Accepted November 09, 2018

Abstract Modern day technology, daily seeks to better data processing activities through features such as improved speed, better functionality, higher mobility, portability and improved data access – all of which is extended via smart computing. The widespread use of smartphone has led to an exponential growth in the volumes of emails, alongside great success in phishing attacks carried out more effective via spam inbox mails to unsuspecting users – soliciting for funds. Many mail apps today, offers automatic filters as a set of rules to help better organize and dispose (as spam, if necessary) incoming mails based through the checking of certain keywords detected in a message's header or body. Achieving such programming filter feature is quite mundane and also inefficient, as spams often evade such filters, slipping into inbox again and again. The study seeks to provide an intelligent adaptive mail support that learns user's preference via an evolutionary unsupervised model(s) as a computational alternative that adapts the data locality feat as well as compares convergence results yielded by the unsupervised hybrid classifiers. It achieves such feats by building local decision heuristics into their classification processes so that such spam filter(s) are embedded with a design that allows for email genres.

Keywords: *evolutionary models, spams, filters, SVM, PHMM, Neural network, classifiers*

Cite This Article: 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 (2018): 9-15. doi: 10.12691/dt-3-1-2.

1. Introduction

SPAM (junk) mails have been defined by many researchers in relation to how they differ from genuine or non-spams. The shortest among these describes spam as 'unsolicited bulk email' [1,2]. It is also 'unsolicited, unwanted email sent indiscriminately, directly or indirectly by a sender with no current relationship to the unsuspecting user' [3]; But, one of the most widely accepted definitions was presented by SpamDefined [4] as 'unsolicited messages, sent or posted as part of a larger collection of messages, having substantially identical content. Spam advertises various goods and services – with dedicated percentage that change over time [5]. This changeability has become a big challenge used by social engineers, especially in the local nature relating to concept drift in spam [6]. This problem has become an imperative concern as spams constitute over 80% of total emails received [7] resulting in direct financial losses through misuse of traffic, storage space, and computational power [8].

Spams normally wastes the processors time, and leads to the loss in productivity and violation of privacy rights. It has also caused several legal issues via pornographic advert, pyramid schemes, Ponzi-schemes etc [9]. The total worldwide financial losses caused by spam estimated by Ferris Research Analyzer Information Service were over $50 billion [10]. Phishing are special cases of spamming activity found to be dangerous and difficult to control – because it particularly hunts for sensitive data (such as passwords, credit card numbers, etc.) by imitating requests from trusted authorities such as banks, server administrators or service providers [11,12]. Social engineering attack is on the rise and this calls for a growing scientific finding(s) to address the characteristics of spamming as well as offer feasible controls.

1.1. E-mail Transfer Protocols and Spam Attacks

These protocols seek to enhance or completely substitute the existing standards of email transmission by new spam-proof variants. A main drawback of the commonly used Simple Mail Transfer Protocol (SMTP) is in the non-reliable mechanism of checking the identity of a message source. But, Sender Policy Framework (SPF) protocol overcomes this issue by inventing more secured method of

packaging the sender's identification. Other variants have been developed to address these issues such as Designated Mailers Protocol, Trusted E-Mail Open Standard, SenderID mechanism etc [13].

Spammers often evolve their techniques to outpace known filter methods and make them ineffective via misclassification of these threats. Through a systematization proposed by Wittel and Wu [14] thus, attacks on spam filters include:

a. Tokenization: If the spammer intends to prevent correct tokenization of the message by splitting or modifying feats such as putting extra spaces in the middle of words.
b. Obfuscation: When contents of the message is obscured from the filter through the process of encoding.
c. Statistical attacks: When the spammer intends to skew the message's statistics. If the data used for a statistical attack is purely random, the attack is called weak. Otherwise it is called strong. An example of strong statistical attack is good word attack as postulated by Daniel and Christopher [9].

1.2. Learning-Based Spam Filtering Methods

Filtering is a popular solution to the problem of spam. It is an automatic classification of messages into genuine and spam mails. Existing filtering algorithms are effective with above 90% accuracy at experimental evaluation. These algorithms can also be applied at the different phases of email transmission such as at routing stage, at the destination mail server, or in the destination mailbox [15]. Filters prevent end-users from wasting their time on junk messages. It does not prevent the misuse of resources as all messages must be delivered. Thus, it has been argued that filtering at the destination only gives a partial solution to spam problems. Figure 1 shows various components of an e-mail that can be analyzed by a filter. To effectively classify new messages, a filter can analyze these components separately (by checking the presence of certain words in case of keyword filtering), or in groups (by considering that the arrival of a dozen of substantially identical messages in five minutes is more suspicious than the arrival of one message with the same content).

An e-mail message typically consists of two parts: (a) a header, and (b) its body. A message body is usually a text in a natural language, possibly with HTML mark-up and graphical elements; while, the header is a structured set of fields, each having a name, value, and specific meaning. Some fields (such as *From*, *To*, *Subject*) are standard; while others depend on the software used in message transfer such as filters installed on mail servers. The subject field contains what a user sees as the subject of message, which is treated as a part of the message body. The body is also the contents of the message; And, non-content features are not limited to the features of the header. For example, a filter may consider the message size as feature (used as a training data, pre-collected messages). This can be optimized as involving users' collaboration to receive multiple user inputs about new messages for analysis.

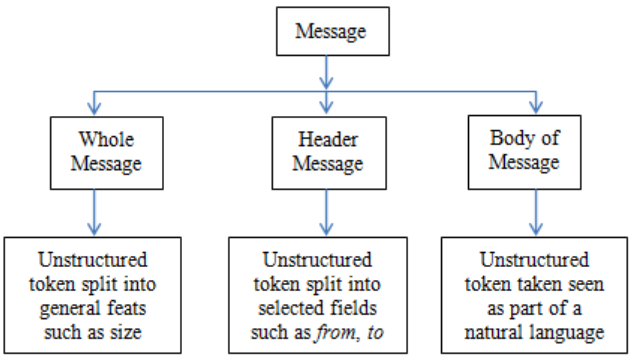

Figure 1. Components of the E-mail

1.3. Statement of Problem / Study Goal(s)

a. Discrete models often yield inconclusive solutions due to noisy inputs. Thus, yields increased rate of false-positive (unclassified) and true-negative (wrongly classified) data. Our hybrid models seek to effectively classify data into distinct classes, compare both results from predictive data mining rules and reinforcement learning (as in Section II).
b. Data as encoded consist of imprecision, ambiguity, noise and impartial truth, easily resolved through robust search (as in Section II with unsupervised models).
c. Parameter(s) selection is a daunting task when seeking for optimized solution in chaotic and dynamic events such as this. Thus, careful selection is required so that model is devoid of over-parameterization, over-fitting and over-training (resolved in Section II) as model seeks to unveil the underlying probability of the data feat(s) of interest.
d. In hybrid, conflicts must be resolved such as: (a) conflict imposed by the underlying statistical dependencies in the adopted heuristics, and (b) conflict imposed in encoding of dataset used. Proposed model resolves this (Section II) via creation of profiles that effectively assigns/classifies data into spams and genuine e-mails.
e. Speed constraints arise in most intelligent (supervised or unsupervised) models as they mainly use hill-climbing, so that model's speed shrinks as it approaches optima. Some use their speed benefits to find optima (at training or testing) to avoid being trapped at local maxima, especially in cases where we seek a single, global rule. However, in spam detection, we seek an optimal solution of several rules good enough to effectively classify dynamic profiles (resolves in Section II) into spams and genuine e-mails.

The goal of this study is to seek for computational intelligence between supervised and unsupervised models, corresponding hybrids, data encoding as model seeks convergence with the input dataset, seeking a robust optima solution guaranteed of high durability and void of noise and conflicts caused by the underlying probability cum statistical dependencies as well as other parameters of interest as we seek the model that is best suited for such dynamic and effective classification (problems) of spam from genuine emails.

1.4. Rationale for Study and Design

Spams are a cheap and illegal form of advert that exploits the electronic mail infrastructure, and easily reach thousands of unsuspecting users. Implementing a reliable spam filters has become imperative – to deal with the growing amount of these uninvited e-mails. Origin or address-based Anti-spam resident at the recipients' end of the mailing infrastructure typically use network data to classify spams; while content filters examines the actual contents of emails. Several mechanisms are in use to address spamming; Each has its shortcomings to make them less effective. Supervised models have been used to solve such problems relating to text classification successfully. Data variation or outliers however, have very negative impact on the classification efficiencies of these two systems.

2. Experimental Models

2.1. Naïve Bayes (Benchmark) Model

This classifier is a simple probabilistic classifier based on Bayes' theorem with strong independence assumptions. A more descriptive term for the underlying probability model is 'independent feature model'. The model assumes that the presence (or lack of presence) of a particular feature of a class is unrelated to the presence (or lack of presence) of any other feature. Despite the fact that the far-reaching independence assumptions are often inaccurate, the naive Bayes classifier has several properties that make it surprisingly useful in practice. In particular, the decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a 1-dimensional distribution. Each message is represented by a vector x = (x1, x2, x3,…, xn) – where x1…xn are the values of attributes X1…Xn. Its binary attributes: Xi = 1 is used if some characteristic represented by Xi is present in message; Else, Xi = 0. In spam filtering, attributes correspond to words, i.e. each attribute shows if a particular word (e.g. "deceased", "died") is present. To select among all possible attributes, we compute the mutual data (MD) of each candidate attribute X with the category denoting the variable C.

2.2. Hybrid HP-SVM-NN (Benchmark) Supervised Models

SVM uses associated learning to analyze data and recognize patterns in classification and regression analysis. It takes a set of data inputs in n-dimensional space, and maps them into one of two classes, creating a separating hyper-plane such that it trains the data cum assigns new data into the classes via a non-probabilistic binary linear classifier [12,16,17]. Each data is represented as a point in the space, and mapped easily into each separate class due to the wide gap between the two classes. To compute the margin, model constructs two parallel hyper-planes so that new data are predicted to belong to a class depending on the class or side of the gap they fall into. It efficiently performs non-linear classification via 'kernel trick' by implicitly mapping inputs into a high-dimensional feature space [18]. We note that the larger the margin – the better the generalization error of the classifier. Classes may overlap since each data is treated as a separate binary classification problem/task [13].

K-nearest neighbour assumes that data points close together are likely to have the same classification. The probability that a point x belongs to a class is estimated by the proportion of training points in a specified neighbourhood of x that belong to that class. The point is either classified by a majority vote (where number of points in neighbourhood belonging to each class is counted, and the class to which the highest proportion of points belongs to is the most likely classification of *x*) or by a similarity degree sum (which calculates a similarity score for each class based on the *K*-nearest points and classifies *x* into class with the highest similarity score). Its lower sensitivity to outliers allows majority voting to be commonly used rather than the similarity degree sum [19]. It uses majority vote to determine which points belongs to the neighbourhood so that distances from *x* to all points in the training set must be calculated. Any distance function that specifies which of two points is closer to sample point is used [20], and the most common distance metric used (in knn) is the Euclidean distance [21] so that each test point f_t and training set f_s, with *n* attributes is calculated via Eq. 1, achieves this via these steps [22]: (a) chosen *k* value, (b) compute distances, (c) sorts the distance, (d) finds *k*-class values, (e) finding dominant class.

$$d = \left[(f_{t1} - f_{s1})^2 + (f_{t2} - f_{s2})^2 \ldots + (f_{tn} - f_{sn})^2 \right]^{\frac{1}{2}}. \quad (1)$$

A major challenge in using *K*nn is to determine optimal size of *k* that acts as smoothing parameter. A small *k* is in sufficient to accurately estimate the population proportions around the test point. A larger *k* will result in less variance in probability estimates (and introduce more bias). Thus, *k* should be large enough to minimize probability of a non-Bayes decision, and small enough that all points give an accurate estimate of true class. Enas and Choi [23] An optimal value of *k* depends on sample size, covariance in each population and the proportions for each population in total sample. If there are differences in covariance matrices, and if the difference between the sample proportions are both small/large, an optimal *k* becomes $N^{3/8}$ (*N* is number of samples in the training set). If there is a large difference between covariance matrices, and small difference between sample proportions, the optimal *k* becomes $N^{2/8}$ [23]. The merits of knn [24] includes: (a) it is mathematically simple, (b) its classification results are good, (c) it is free of statistical assumptions, (d) its effectiveness does not depend on the space distribution of classes, and (e) if boundaries between classes are not hyper-linear or hyper-conic, Knn outperforms LDA.

Okesola et al [25] the hybrid features data locality in spam filtering. Spam is not uniform; But, consist data (messages) on different topics [26] and with different genres [16]. Accuracy is improved if classification by the model is based on local decision rules. Thus, it uses SVM

to provide a global decision rule independent of sample which must ordinarily be classified. Accurate classification of spam is limited to the fact that spam consist of messages on various topics and genres. Local decision rules must thus, be applied in collaborative filtering as opposed to the present application of global rules that sees and classifies spam based on pre-coded data on genre and types. The changeability of data is also likely to have local nature [6], and this is applicable to genuine mails also. The existence of algorithms which classifies email by topic provides the evidence of both locality in genuine mail and the possibility to capture it using bag-of-words feature extraction. The simplest filtering method that engages data locality uses knn model and it has been found to outperform SVM on spam classification [27]. This suggests that a more elaborate way of building local decision rules is highly required.

Okesola et al [25] has these errors (implicitly stated as):

a. They noted that spam(s) is not uniform. How can a single (global) rule effectively classify such dynamic message (and its contents)?
b. How did the model encode datasets used in the hybrid especially for such dynamic data (message changes) that is rippled with ambiguities, noise and impartial truth?
c. Parameter selection at training and testing was not clearly stated and number of runs that result in their convergence as we seeks to discover underlying probability of the data feat(s) of interest.
d. How did they resolve conflicts imposed by the underlying statistical dependencies in the adopted heuristics as well as that imposed in encoding of the dataset used?
e. What speed constraints were experienced?
f. Because supervised models yield inconclusive solutions of *unclassified* (false-positive) data and wrongly *classified* (true-negative) data. What improvements are experienced by the model and at what rate (with its predictive data-mining rules and reinforcement learning)?

2.3. The Genetic Algorithm Trained Neural Network (GANN)

Ojugo et al [28] described the genetic algorithm trained neural network as adapted in the early detection of diabetes. GANN is initialized with (n-r!) individual if-then, fuzzy rules (i.e. 6-4!). Individual fitness is computed as 30-individuals are selected via the *tournament* method to determine new pool and selection for mating. Crossover and mutation is applied to help ***net*** learn the dynamic, non-linear underlying feats of interest via multipoint crossover to yield new parents. The new parents contribute to yield new individuals. Mutation is reapplied and individuals are allotted new random values that still conform to the belief space. The mutation applied depends on how far CGA is progressed on the net and how fit the fittest individual in the pool (i.e. fitness of the fittest individual divided by 2). New individuals replace old with low fitness so as to create a new pool. Process continues until individual with a fitness value of 0 (i.e. solution) is found [28,29].

Table 1. Fuzzy Encoded Class

Code	Fuzzy Parameters	Genuine	Spam
P01	Message size	0.50	0.50
P02	Message Attachment	0.50	0.50
P03	Header From	0.50	0.50
P04	Header To	0.50	0.50
P05	Header Subject	0.30	0.70
P06	Body of Message	0.25	0.75

Fitness function (*f*) is resolved with initial pool (Parents) using the genuine class as thus:
R1:50 R2:50 R3:50 R4:50 R5:30 R6:50.

Table 2. 1st and 2nd Generation of population from Parents

S/N	Selection	Chromosomes (Binary 0 or 1)			Fitness Function
		Parent 1st Gen	Crossover	Parent 2nd Gen	
1	50	110010	1 and 6	110**01**	49
2	50	110010	2 and 5	110**10**	50
3	50	110010	3 and 6	110**01**	49
4	50	110010	4 and 5	110**10**	50
5	30	011110	5 and 6	011**01**	29
6	25	011001	6 and 5	011**10**	26

Initialization/selection via ANN ensures that first 3-beliefs are met; mutation ensures fourth belief is met. Its influence function influences how many mutations take place, and the knowledge of solution (how close its solution is) has direct impact on how algorithm is processed. Algorithm stops when best individual has fitness of 0.3 [30]. Model stops if stop criterion is met. GANN utilizes number of epochs to determine stop criterion.

2.4. The Profile Hidden Markov Model (PHMM)

Ojugo et al [18] describes the Hidden Markov model as used in examination scheduling. Adapted to spam detection classification problem, probability from one transition state to another is as in Figure 2. In spam detection analysis, a rule not accepted by the trained HMM, yields high probability of either a false-positive or true-negative [18]. Our study adopts the Profile HMM (a variant of HMM), which offers the following merits to the fundamental problems of the HMM by: (a) makes explicit use of positional (alignment) data contained in observations/sequences, and (b) allows null transitions (if necessary) so that model can match sequences that includes insertion and deletions [31].

Traditional HMM scores data via clustering based on profile values. Our PHMM samples probabilities of initial set of rules, and classify them into spam and genuine class. It then stores a log in memory to reduce high true-negative results (i.e. rules with semblance of spam) and high-false positives (unclassified rules). Thus, the model is initially trained to assimilate normal behaviour of both classes. It then creates a profile of the rules, classifying them into genuine and spam profiles [18].

In spam detection, O is each rule contained to define various feats of a spam, T is time it takes each rule to classify a data input, N is number of unclassified/wrongly classified rules, M is number of rules accurately classified, π is the initial state (starting rule), A is state transition probability matrix, a_{ij} is probability of a transition from a state i to another state j, B contains N probability distributions for the codes in the knowledgebase where profiles have been created (one rule for each state process), and HMM λ = (A,B,π). Parameters for HMM details are incomplete as above; But, the general idea is still intact [18]. Figure 2 shows G is *genuine* rule class, S is the *spam* rule-class, UC is *unclassified* rules, and WC is wrongly *classified* class.

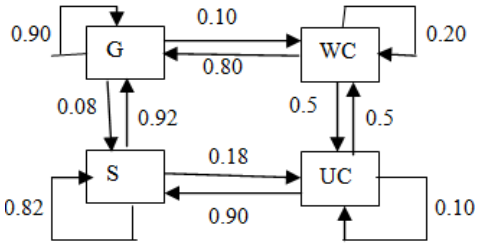

Figure 2. Actual State Transition with P(x)

We align multiple (data) rules as sequence with significant relations. The output sequence determines if an unknown code is related to sequence belonging to either of the genuine class or spam (those not contained in the Bayesian net). We use the PHMM to score rules and to make decision(s). Circles are ***delete*** state that detects rules as classified into the mail types, rectangle are ***insert*** states that allows us to ***accurately*** classify rules that have been previously unclassified/wrongly classified into a class type (and consequently, update the knowledgebase of classified false-positives and true-negatives); diamonds are ***matched*** states that accurately classifies rules as in standard HMM [18,31]. The delete and insert are emission states in which observation is made as PHMM passes through all the states. Emission probabilities, corresponding to B in standard HMM model is computed based on frequency of symbols that can be emitted at a particular state in the model; However, they are positional-dependent (in contrast to basic HMM). The emission probabilities are derived from Bayesian net, which represents our training phase. Finally, ***match*** states allow the model to pass through gaps, existing in the Bayesian net to reach other emission states. These gaps prevent model from over-fitting and overtraining [18]. And use the forward algorithm recursively computes probabilities of all possible case by reusing scores calculated for partial sequences using Eq. 2 to Eq. 4 respectively as thus:

$$F_j^M = Log\frac{eM_j(x_i)}{qx_i} + \log(aM_{j-1}M_j \exp\left(F_{j-1}^M(i-1)\right) + aI_{j-1}M_j \exp\left(F_{j-1}^I(i-1)\right) + aD_{j-1}M_j \exp\left(F_{j-1}^D(i-1)\right) \quad (2)$$

$$F_j^I = Log\frac{eI_j(x_i)}{qx_i} + \log(aM_jI_j \exp\left(F_j^M(i-1)\right) + aI_jI_j \exp\left(F_j^I(i-1)\right) + aD_jI_j \exp\left(F_j^D(i-1)\right) \quad (3)$$

$$F_j^D = \log(aM_{j-1}D_j \exp\left(F_{j-1}^M(i)\right) + aI_{j-1}D_j \exp\left(F_{j-1}^I(i)\right) + aD_{j-1}D_j \exp\left(F_{j-1}^D(i)\right) \quad (4)$$

3. Result Findings and Discussion

3.1. Findings and Discussion

To measure their effectiveness and classification accuracy, we measure their rate of misclassification and corresponding improvement percentages in both training and test data sets as summarized in Table 3 and Table 4 respectively. The equations for the misclassification rate and its improvement percentage of the unsupervised (B) model against those of the supervised (A) model, is respectively calculated as follows:

$$Misclassification\ Rate(MR) = \frac{No.of\ Incorrect\ Diagnosis}{No.of\ Sample\ set}. \quad (5)$$

Table 3. Misclassification Rate of Each model

Model	*Classification Errors*	
	Training Data	Testing Data
Naïve Bayes	52.5%	45.2%
HP-SVM-NN	48.4%	33.7%
PHMM	19.6%	13.2%
GANN	22.5%	22.01%

Also, its improvement percentage is computed as thus:

$$Improvement\ Percentage = \frac{MR(A) - MR(B)}{MR(A)} x100 \quad (6)$$

Table 4. Improvement Percentage

Model	*Improvement %*	
	Training Data	Testing Data
Naïve Bayes	10.1%	15.6%
HP-SVM-NN	26.67%	29.02%
PHMM	56.03%	64.16%
GANN	42.79%	34.09%

Results obtained from Table 3 and Table 4 respectively shows that unsupervised (PHMM/GANN) model outperforms supervised (Naïve Bayes and hybrid HPSVMNN) models. PHMM has a rate of misclassification of 13.2% (i.e. low error rates in false-positives and true-negatives classes). Consequently, it has a classification accuracy of 87%; It promises improvement rate of about 64.16%. Conversely, our memetic algorithm (GANN) has a misclassification rate of 22.01% (of false-positives and true-negatives error rate); while it promises to improve by 34.1%. Hybrid HP-SVM-NN has a rate of misclassification of 33.7% (i.e. for false-positives and true-negatives error rates). That is, it shows classification accuracy of 67%; While, promising an improvement rate of about 29%.

3.2. Related Literature

Longe et al [13] developed SPAMAng, a Naïve Bayesian System for outbound e-mails using a support vector machine, open-source implementation. Its result indicates that for both systems (using a set of carefully selected fraudulent e-mails) that an outlier introduces vulnerability into SVMs – causing it to be defeated by spammers. SVMs performance degradation is noticeable when used with fraudulent spams filtering (419 mails) where spammers engage in concept drifts using text manipulations, phishing and spoofing, to fool spam filters. The comparison of SVMs with SPAMAng showed that SVMs does not always produce best result in all text classifications.

Barakat et al [32] used SVM with additional intelligent module to transform the SVM black-box to an intelligent diagnostic model with adaptive results that provides potential model for diabetes prediction. Its logical rule set generated had prediction accuracy of 94%, sensitivity of 93%, and specificity of 94%. Extracted rules are medically sound and agree with outcome of relevant medical studies.

Khan et al [33] used the fuzzy trained neural network in evaluating the well-known Wessinger's quadratic function as a constraints satisfaction problem. His results show that the model was able to bring closer the data points in the range of the analytical solution; while, Khasei et al [34] used a feed-forward MLP. It can be expanded and extended to represent complex dynamic patterns or cases such as this, since it treats all data as new – so that previous data signals do not help to identify data feats of interest, even if such observed datasets exhibits temporal dependence. However, it is more practically difficult to implement as the network gets larger.

4. Recommendations/Conclusion

4.1. Rationale of the Study

The rationale for the model choice is to compare between: (a) clustering profile versus hill-climbing, and (b) convergence behavior and other statistic. The PHMM converged after 253-iterations; GANN converged after 213-iterations; while hybrid HP-SVM-NN converged after 459-iterations. PHMM perform best in cluster (profiling) tasks where we seek various rules that can effectively classify items. GANN is better than PHMM in some other tasks. We ***note***, model's speed is traded-off for greater accuracy of classification, more number of rule set generated to update the knowledge database for optimality and greater functionality. Also, ***Jordan*** net overcomes such difficulty [34] via the use of its internal feedbacks that also makes it appropriately suitable for such dynamic, non-linear and complex tasks as its output unit is fed-back as input into its hidden unit with a time delay, so that its outputs at time $t-1$, is also input at time t.

4.2. Conclusion

Hybrids are difficult to implement and its accompanying data must be appropriately encoded so that model can exploit numeric data and efficiently explore the domain space to yield an optimal solution. Modelers must seek proper parameter selection and adjustment of weights/biases so as to avoid ***over-fitting***, ***over-training*** and ***over-parameterization*** of the model. Encoded through the model's structured learning, this will help address issues of statistical dependencies between the various heuristics used, highlight implications of such a multi-agent populated model as well as resolve conflicts in data feats of interest. Thus, as agents create/enforce their own behavioral rules on the dataset, hybridization should be able to curb this (as CGA does for this model in its belief space and operators as applied) to display the underlying probabilities of interest.

Models serve as educational tools to compile knowledge about a task, serve as new language to convey ideas as we gain better insight to investigate input parameter(s) crucial to a task; while, its sensitivity analysis helps to reflect on theories of systems functioning. Simple model may not yield enough data; and, complex model may not be fully understood. A detailed model helps us develop reasonably-applicable models even when not operationally applicable in a larger scale. Their implementation should seek its feedback as more critical rather than seeking an accurate agreement with historic data. Since, a balance in the model's complexity will help its being understood and its manageability, so that the model can be fully explored as seen here [35].

References

[1] Androutsopoulos, I., Koutsias, J., Konstantinos V and Constantine, D., (2005). *An experimental comparison of naïve bayesian and keyword-based anti-spam filtering with personal e-mail messages*, Proc. of 23rd annual ACM SIGIR Conf. on research and development in information retrieval, SIGIR'00, pp. 160-167.

[2] SPAMHAUS (2005). The definition of spam. Available online at http://www.spamhaus.org/definition.html.

[3] Cormack, G., and Lynam, T. (2005). Spam corpus creation for TREC. In Proceedings of Second Conference on Email and Anti-Spam, CEAS'2005. http://ceas.cc/2005/.

[4] Spam Defined. (2001). Spam defined, Online: www.monkeys.com/spamdefined.html.

[5] Lorenzo, L., Mari, M. and Poggi, A. (2005). Cafe – collaborative agents for filtering e-mails. In Proceedings of 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise, WETICE'05, pages 356-361.

[6] Delany, S., Padraig, C., Alexey, T and Lorcan, C., (2004). *A case-based technique for tracking concept drift in spam filtering*, Knowledge-based systems, pp 187-195.

[7] MAAWG (2006). Messaging anti-abuse working group, Email metrics report for third & fourth quarter 2006, Online at: www.maawg.org/about/MAAWGMetric200634report.pdf.

[8] Mikko, S and Carl, S., (2006). *Effective anti-spam strategies in companies: An international study*, Proc. of HICSS '06, Vol. 6

[9] Daniel, L and Christopher, M., (2005). *Good word attacks on statistical spam filters*, Proc. of Second Conference on Email and Anti-Spam, CEAS'2005.

[10] Ferris Research (2015). *The global economic impact of spam*, report #409. http://www.ferris.com/get content file.php?id=364.

[11] Christine, D., Oliver, J. and Koontz, E. (2004). Anatomy of a phishing email. In Proceedings of the First Conference on Email and Anti-Spam, CEAS'2004

[12] Ojugo, A.A and Eboka, A.O., (2014). *An intelligent hunting profile for evolvable metamorphic malware*, African Journal of Computing and ICT, Vol. 8, No. 1, Issue 2, pp 181-190.

[13] Longe, O.B., Robert, A.B.C., Chiemeke, S.C and Ojo. F.O., (2008). *Feature Outliers And Their Effects On The Efficiencies Of Text Classifiers In The Domain Of Electronic Mail*, The Journal of Computer Science and Its Applications, 15(2).

[14] Wittel, G. and Wu, F. (2004). On attacking statistical spam filters. In Proceedings of First Conference on Email and Anti-Spam, CEAS'2004.

[15] Agarwal, R., Aggarwal, C and Prasad, V., (2001). *A tree projection algorithm for generation of frequent itemsets*, Journal of Parallel and Distributed Computing, pp350-371.

[16] Cukier W., Cody, S and Nesselroth, E. (2006). Genres of spam: Expectations and deceptions, Proc. of the 39th Annual Hawaii International Conference on System Sciences, Vol. 3. www.computer.org/csdl/proceedings/hicss/2006/2507/03/250730051a.pdf.

[17] Ojugo, A.A., (2015). *A comparative stochastic model solution on convergence problem of quadratic functions*, Unpublished Thesis, Mathematics and Computer Science Department, Federal University of Petroleum Resources Effurun.

[18] Ojugo, A.A., Allenotor, D and Eboka, A.O., (2016). *Solving for convergence solution and properties of quadratic function: A case of selected intelligent supervised models*, FUPRE Technical Report (TRON-119), pp 34-42.

[19] Chaovalitwongse, W., (2007). *On time series k-nearest neighbor classification of abnormal brain activity*, IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, Vol. 37.

[20] Fix, E. and Hodges, J., "Discriminatory analysis -Nonparametric discrimination: Consistency properties", Project No. 2-49-004, Report No. 4, Contract No. AF 41(128)-31, USAF School of Aviation, Randolph Field, Texas, 1951.

[21] Viaene, S., Derrig, R., Baesens, B., and Dadene, G., (2002). *A comparison of state - of - the art classification techniques for expert automobile insurance claim fraud detection*, The Journal of Risk and Insurance, Vol. 69, pp. 373-421.

[22] Yildiz, T., Yildirim, S., Altilar, D., (2008). *Spam filtering with parallelized KNN algorithm*, Akademik Bilisim.

[23] Enas, G. and Choi, S., (1986). *Choice of the smoothing parameter and efficiency of k-nearest neighbor*, Computers and Mathematics with Applications, Vol. 12, pp. 235-244.

[24] Berrueta, L., Alonso-Salces, R., Heberger, K., "Supervised pattern recognition in food analysis", Journal of Chromatography A, 1158, pp. 196-214, 2007.

[25] Okesola, J.O., Ojo., F.O., Adigun, A.A and Longe, O.B., (2015). *Adaptive high probability algorithms for filtering advance fee fraud emails using the concept of data locality*, Computing, Information Systems, Development Informatics and Allied Research Journal, Vol. 6, No. 1, pp 7-12.

[26] Hulten, G., Penta, A., Gopalakrishnan, S. and Manav, M. (2004). Trends in spam products and method, Proceedings of the 1st Conf. on Email and Anti-Spam, CEAS'2004, 2004.

[27] Lai, C. and Tsai, M. (2004). An empirical performance comparison of machine learning methods for spam e-mail categorization. Hybrid Intelligent Systems, pages 44-48, 2004.

[28] Ojugo, A.A., Allenotor, D., Oyemade, D.A., Longe, O.B and Anujeonye, C.N., (2015). *Comparative stochastic study for credit-card fraud detection models*, African Journal of Computing & ICTs. Vol. 8, No. 1, Issue 1. Pp 15-24.

[29] Ojugo, A.A., J. Emudianughe., R. Yoro., E. Okonta., A. Eboka., (2013). *A hybrid neural network gravitational search algorithm for rainfall runoff modeling and simulation in hydrology,* Progress in Intelligence Computing and Applications, 2(1): 22-33.

[30] Dawson, C and Wilby, R., (2001). *Comparison of neural networks in river flow forecasting*, J. of Hydrology and Earth Science, SRef-ID: 1607-7938/hess/2001-3-529.

[31] Ojugo, A.A., Ben-Iwhiwhu, E., Kekeje. O., Yerokun, M., Iyawah, I.J.B., (2014). *Malware propagation on social time varying networks: a comparative study of machine learning frameworks,* International Journal of Modern Education Computer Science, 6(8): pp25-33.

[32] Barakat, N.H., Bradley, A.P and Barakat, M.N., (2010). *Intelligible support vector machines for diagnosis of diabetes mellitus*, IEEE Transactions on Information Technology in Biomedicine, 14(4), pp1114-1120.

[33] Khan, J., Zahoor, R and Qureshi, I.R., (2009). *Swarm intelligence for problem of non-linear ordinary differential equations and its application to Wessinger equation*, European Journal of Science Research, 34(4), pp. 514-525.

[34] Khashei, M., Eftekhari, S and Parvizian, J (2012). *Diagnosing diabetes type-II using a soft intelligent binary classifier model,* Review of Bioinformatics and Biometrics, 1(1), pp 9-23.

[35] Ojugo, A.A., Eboka., A., E Okonta., R. Yoro., F. Aghware., (2012). *Genetic algorithm rule-based intrusion detection system (GAIDS)*, Journal of Emerging Trends in Computing Information System, 3(8): pp1182-1194.

[36] Andrew Farrugia (2004). Investigation of Support Vector Machines for Email Classification. Dissertation submitted to the School of Computer Science and Software Engineering Monash University.

[37] Blanzieri, E and Bryl, A., (2007). *Highest Probability SVM Nearest Neighbor Classifier For Spam Filtering*, March 2007 Technical Report DIT-07-007, retrieved on January 2017.

[38] Zhou, F., Zhuang, L., Zhao, B. Huang, L., Joseph, A. and Kubiatowicz, J. (2003). Approximate object location and spam filtering on peer-to-peer systems. In Proceedings of ACM/IFIP/USENIX International Middleware Conference, Middleware.