Multiagent Based Spam Filtering System

***1 Assist. Lect. Dalia ShihabAhmad, *2 Assist. Prof. Dr.MahaBayati**

***1 Assist. Lect. /Computer Science Dept./Faculty of Science/ Mustansiryha University / Baghdad, Iraq.dalia_shihab@uomustansiriyah.edu.iq ,**

***2 Assist. Prof. /Computer Science Dept./Faculty of Science/ Mustansiryha University / Baghdad, mahabayati@uomustansiriyah.edu.iq**

Abstract: The rapid spread and the easy availability of a free e-mail service have made it the medium of choice for the sending of unsolicited advertising and bulk e-mail in general. These messages are called spam e-mail and are an increasing problem to both Internet users and Internet Service Providers (ISPs). In this paper, a multi-agent based architecture is adopted to allow for efficient on-line application of the developed filter while maintain the scalable design that lend itself to openness, a multi-agent system (MAS) designed to implement a spam filter system based on mobile agent technology with SVM-filter that works background, at client side, to improve classification accuracy of Gmail and to allow for distant classification.

Keywords: Multiagent, spam filter, ISPs, SVM

1. Introduction

Spam seems to be a growing problem. Since e-mail becomes the effective communication tools for internet users because it is free, fast and provides easy communication for various purposes (Singh & Bhardwaj, 2018). Hence, is important to avoid spams by apply an efficient spam e-mail filter to categorize an incoming e-mail as non-spam or spam e-mail, and to remove those only spam e-mails (**Teli&Biradar, 2014**). SVMs were considered to efficiently detect spam e-mails. SVM is a statistical approach which becomes popular so quickly due to its rapid way of processing and accurate results. It has ability to deal with high-dimensional data and number of tuning parameters. SVM defines a hyper-plane to divide the n-dimensional space (i.e., n-featured) sample of data into two regions (classes). Each region contains only the training data samples belonging to either class. Many possible hyper-planes can be defined. The SVM objective is to find the one that with the maximum margin, i.e., the maximum distance between data sample of the two classes. A margin is calculated by finding the distance between a hyper-plane and the nearest-sample of the two classes. This "maximums margin hyper-plane" aspect, be informs confident classification power of SVM for future data (**Abdul, 2011)(Kolluru, 2013**). In order to build a robust, scalable and reusable distributed system it is necessary to employ techniques of decomposition and abstraction. Agent technology is suited for constructing such software systems **(Vyroubal&Kušek, 2013**). The term "agent", is defined as a specific program entity that performs some set of tasks on behalf of users with some degree of autonomy. An agent system can base on a solitary agent working within an environment and if necessary, interacting with its users, in which case, multiple agents are needed, such Multi-Agent Systems (MAS) can model complex systems and introduce the possibility of agents having common or conflict goals. These agents may interact with each other both indirectly (by acting on the environment) or directly (via communication and negotiation) **(Hussain, 2014)**. Agent-based systems are adaptable. That is, agents can be modified with new capabilities, and can added or deleted to or from the system on need. Agents can be classified into two distinct categories, static agents and mobile agents. Static agents are software entities that never leave the host on which they were created throughout their lives, while mobile agents can migrate through interconnected hosts (**Vyroubal&Kušek, 2013**). Furthermore, mobile agent is viewed as an entity that can move to a network host connected to execute particular tasks on behalf of the user. During the execution, mobile agents can be created and sent to the destination systems dynamically to implement tasks. The mobility service allows mobile agent to migrate in the network from one host to another, hence providing a number of applications with a flexibility and adaptability necessary to meet the requirements and conditions in a distributed environment (**Liten, 2012**). This work adopts the agent-based architecture to present a MAS. The proposed MAS is designed to implement a distributed spam filter system based on mobile agent technology. A mobile SVM-filter is developed to identify e-mail messages, based on the content (i.e., body) of a message, to filter out those sample e-mail at client side. two datasets of e-mail messages are used to assist classification, and Java Agent Development Environment (JADE) is used to implement the proposed Multi-Agent based Spam Filtering (MASF) System.

2.Related Work

This section presents a number of approaches of filter techniques. To overcome the problem caused by spam many spam filtering solutions were proposed in the recent past years.

Diale, Walt, Celik, & Modupe, (2016),the authors presented spam detection technique using different algorithms such as SVMs, AdaBoost and Random Forests (RF). The SVM classifier was tested against different kernel type and kernel parameters. Experiments were conducted for each classifier and with the use of Enron dataset, SVM showed accuracy of 97.89%, whereas AdaBoost gave accuracy of 97.03% and RF gave accuracy of 97.94%.**Agarwal& Kumar, (2016)**, authors presented a spam filtering system using SVM algorithm with different parameters and different kernel functions (linear, polynomial, RBF, sigmoid). Experiments were conducted for each kernel function and the use of spambase dataset, linear kernel gave accuracy of 92.4381% and outperformed the other kernels.**Shajideen, (2018**), the author presented spam filtering system using SVM, NB and J48 algorithms. A comparative study of these algorithms was performed using enron1, which was taken from the Enron dataset. Experiments showed that, SVM scored 94.06%, whereas NB scored 92.8% and J48 scored 92.07%. Consequently, SVM was approved to be the best classifier in terms of accuracy and False Positive Rate (FPR) despite the time required for training.**Balakrishnan & Shunmuganathan, (2015**), authors considered the concept of collaborative, personalized agent in building Bayesian spam filter as an agent function developed in JADE. The filter consisted of a set of graphical tools to support the performance of filtering phases. Experiments aimed at handling the problem of misclassify those non-spam e-mails. Reclassification was needed upon routing spam e-mails to quarantine file, hence reducing false positive rate (FP).**Jafar & Edalatirad, (2015),** the authors presented a spam filtering system used heterogeneous multi-agent filters where each agent it's own ability to detect. Methods as: NB, DT and neural network are used for learning by agents. This is accomplished via two phases. The first phase depends on the prior knowledge of the agent without control or participation of users. The second phase depends on user's monitoring based on properties of both spam and non-spam e-mail.**Mohammed, Shamini, Mostafa, & Ghani, (2018),** the authorsconstructed multi-agent based spam filtering model called Multi-Natural Language Anti-Spam (MNLAS) model, implemented in a Java environment using JADE agent platform. The MNLAS model comprises of five agents, each agent executed certain task with in the overall process of spam detection. Agent applied RF algorithm to classify spam from non-spam e-mails on the bases of using a dataset of 200 e-mails. Experiments were conducted using visual as well as textual information to be extracted from set of messages to classify the received message as being spam or else, RF showed an average accuracy of 91.90%. **Flaih & Shukur, (2018)**, authors designed and implemented a complete system for switching and filtering messages to solve the problem of spam using software agents and rule-based system. The proposed system is able to check all incoming and outgoing e-mail to block unwanted massages or to replace specific undesired words from other message before sending or receiving them. The system creates a dynamic number of agents according to user's desire in a simply and interactive way. Each agent can have a dynamic number of filtering rules that meet the requirements. Experiments showed that this system differs from others in that being able to display filtered messages after filtering process hiding unsuitable words from filtered messages.

It is worth to know, that no extra work concerning MAS-based mobile filtering system is noticed with in the recent research efforts.

3. System Architecture and Workflow

To best present the design of MASF system, it is worth to notice the modes. This system being through during the course of action "off-line" and "on-line" modes, brining into mind that system's function will be accomplished via a number of cooperative agents, each of which has a clear cut actions to take in it's environment, where this environment strictly sticks to the mode of work just mentioned.In short, the "off-line" is the mode where system establishes the learning model of an SVM-filter on the basis of applying. Training/testing phases using the CSDMC2010 dataset of raw e-mails and a specific setting of corresponding tuning parameters. Just then the system runs in the on-line mode using real data samples are collected manually. To apply the SVM-filter mounted on a server to filter out those incoming spam only e-mails.Figure (1) illustrate the abstract architecture of the proposed system.

3.1 System Component

3.1.1 Off-Line Classification Agent

As shown in Figure (1), this agent actually consists of the three agents: preprocessing agent, feature extraction agent and content-filter agent (all of which works in off-line mode). It aims at establishing a learned classifier model by stepping through the process assigned to each of its constituent agents.

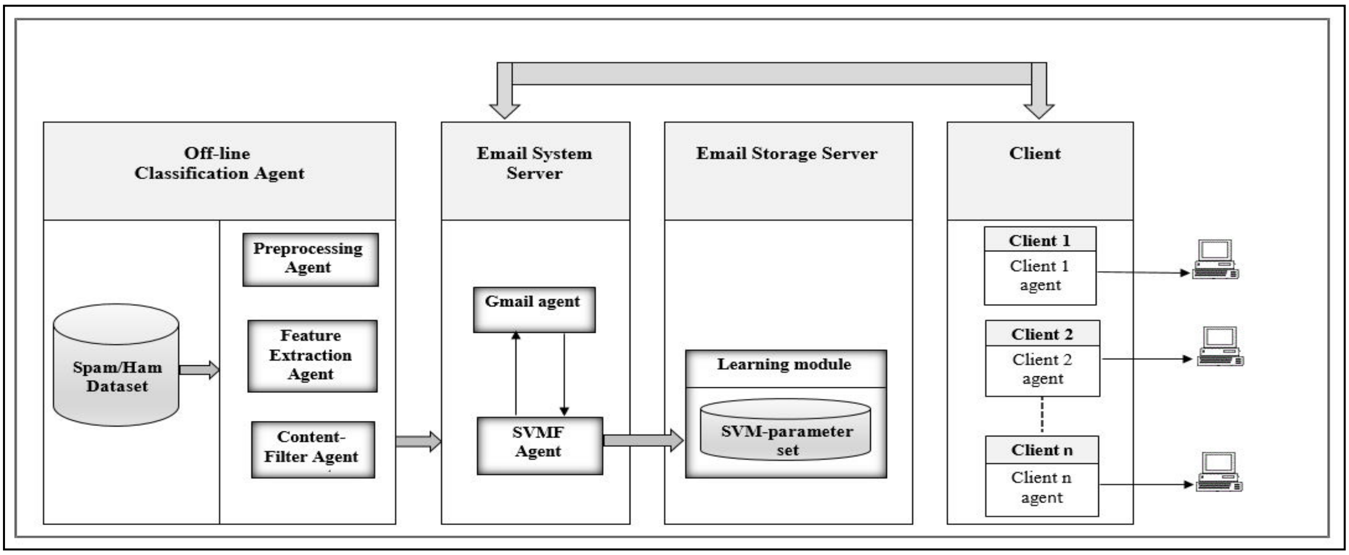


Figure 1: Abstract Architecture of MASF System

A. Preprocessing Agent (PPA)

This is the agent responsible for preprocessing individual e-mails in the dataset. It is function passes through the following phases:

i. Reading .eml file: Read e-mail to extract the body; content of message.

ii. Preprocessing Phase: Preprocess the content of e-mail messages by removing redundant data and keeping only useful information to help classify the e-mail. This phase consists of four steps: Normalization, Tokenization, Stop-word removal and Word Stemming.

B. Features Extraction Agent (FEA)

This agent prepares for building feature space. To accomplish its task, this agent passes through the following phases:

i. Features Extraction Phase: Extract the features from each e-mail's body and represented as vectors in the "Bag of Words (BOW)".

ii. Features Scaling Phase: standardizing the feature value to fall in certain rang. In this work, min-max normalization is used to scale extracted feature values to be in the range [0-1].

C. Content Filter Agent (CFA)

This is the agent responsible for classifying e-mails using SVM algorithm that proceeds through the two phases of training and testing as follows:

i. Training phase: Training an SVM is a long process requiring a large amount of data and many experiments to achieve acceptable result. In this work a training set of labelled e-mails is input to start training through the following sequence of steps:

iii. Testing phase: Categorizing unlabelled e-mails in the data set into spam or non-spam e-mails on the basis of the outcome from training.

3.1.2 The E-mail System Server

This is a multi-agent system that consist of Gmail agent and Support Vector Machine Filter agent (SVMF). It is the core of the on-line manipulation of e-mail provided by propose system.

A. Gmail Agent

This agent allow connection to the Gmail server, have provide access to Gmail filter and Gmail storage. Gmail filter have ability to automatically filtering incoming Gmail message on the Gmail server. Gmail storage is portioned mainly into:

1.Space for incoming unconfirmed e-mail messages.

2.Space for confirmed (classified) e-mail messages, which in turn is portioned into Inbox and Spam box.

B. SVMF Agent

In principle, this agent coordinates the working of the two agents: static agent, namely Mail Check Agent (MCA), and mobile agent, namely Mobile Filter Agent (MFA).

i. Mail Check Agent (MCA)

This agent performs the follow tasks:

1.Frequently checks the e-mail messages in the Gmail confirmed box.

2.Create a mobile agent (MFA) upon arrival of each new e-mail message.

ii. Mobile Filter Agent (MFA)

This is the vital agent among all of the others. It is function is to filter the confirmed message arrived at some client. To do so, MFA moves to the client side to perform "on-line" filtering process on the incoming e-mail. To allow for mobility, MFA need to deal with SVM learn module together with SVM-parameter. MFA goes through the following phases:

1. Reading .eml file.
2. Preprocessing phase.
3. Feature Extracting phase.
4. Testing phase.

These phases function the same ways as presented earlier.

The Communication between agents is accomplished via messages. Table (1) summarizes the messages needed for communication between MCA and MFA.

Table (1): Summarizes Different Types of Messages

Message	Source, Destination Agent	Description	Parameter
REQUEST	MCA, MFA	Request to move MFA to client side	client address, e-mail message
INFORM	MFA, MCA	Confirm movement of MFA	-------------

3.1.3 The E-mail Storage Server

To provide for availability of the learned module, the server hold; the SVM parameter-set needed by MFA to do filter at the client side.

3.1.4 Client Agent

This should be included at each client connected to the system. In this case, MFA do the classification needed to filter e-mail sent to that client and to manipulate that client's inbox and the spam box accordingly.

3.2 Workflow in On-Line Mode

To provide a clear-cut view of the way MAFS system works, we attempted to put together all the agents established there and to use the AUML sequence diagram, shown in Figure (2) to illustrate there working interaction during the course of running of the system which is initiated upon receiving e-mail message by some client.

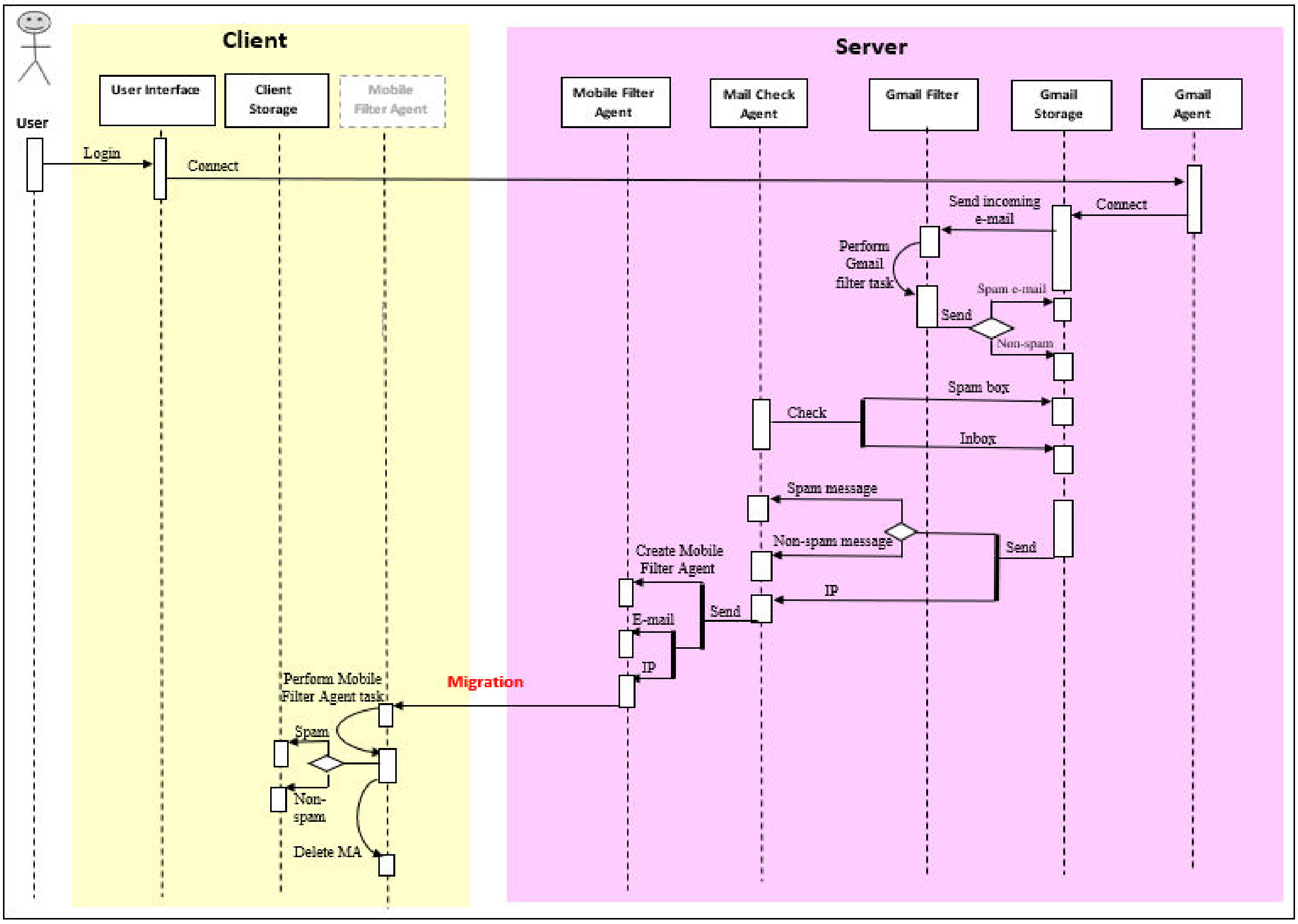


Figure (2): AUML sequence diagram

1. Implementation and Experimental Results

4.1 JADE Perspectives

In order to prove the viability of the proposed protocol, a prototype of multi-agent system is created. The system implementation is made with in JADE agent platform. In JADE, agent instances such as PPA, FEA, CFA, MCA and MFA can be designed by extending the Jade.core.Agent class. Then, each task and communication performed by the agents can be defined using sub-classes of the Jade.core.behaviour class. Tasks accomplished by the system is presented in terms of the mode of work the system in and the relation agent doing that task.

A. Off-line classification mode:

i. Preprocessing Agent (PPA): The task associated with this agent is implemented as a one shot behaviour which consists of two operations, ReadMail() and Prprocessing().

ii. Feature Extraction Agent (FEA): The task associated with this agent is implemented as a one shot behaviour which consists of two operations, FeatureExtraction() and FeatureScaling().

iii. Content-Filter Agent (CFA): The task associated with this agent is implemented as a one shot behaviour which consists of three operations, Training (), Testing () and Validat().

B. On-line classification mode:

The overall work at this mode is carried out by the SVMF agent. Who initiates the following agents:

i. Mail Check Agent (MCA): The task associated with this agent is implemented as a cyclic behaviour which consists of two basic operations, Send Request Create and Move MFA message (RM) and Receive Inform Confirm Movement MFA (IM) message.

ii. Mobile Filter Agent (MFA): Due to its distinguishable feature of "Mobility", this agent is assigning types of behaviour depending on the side of the system this agent is running at:

1. Cyclic behaviour at Gmail server side:prior to migration this agent performs the follow:

- Receive Request Create and Move MFA (RM) message.
- Send Inform Confirm Movement MFA (IM) message.

2. One shot behaviour at client side: Upon agent's migration, this agent perform the four operations *ReadMail(), Prprocessing(), FeatureExtraction()* and *Testing().*

It's worth to mention that, cyclic behaviour and one-shot behaviour are subclasses of the simple class behaviour. Behaviour class and related operations are shown in Figure (3).

As Figure (3) presents the low-level view of JADE implementation, where Figure (4) presents high-level view implementation. At the Server side, the main container in the JADE platform contains the local agents MCA, MFA together with RMA, DF, AMS and AMM (the default agents). These agents communicate with each other through message transport protocol MTP. At the Client side, the MFA, after migration, resides in the main container. Inter Platform Mobility Service IPMS is used to enable the agent mobility to client side.

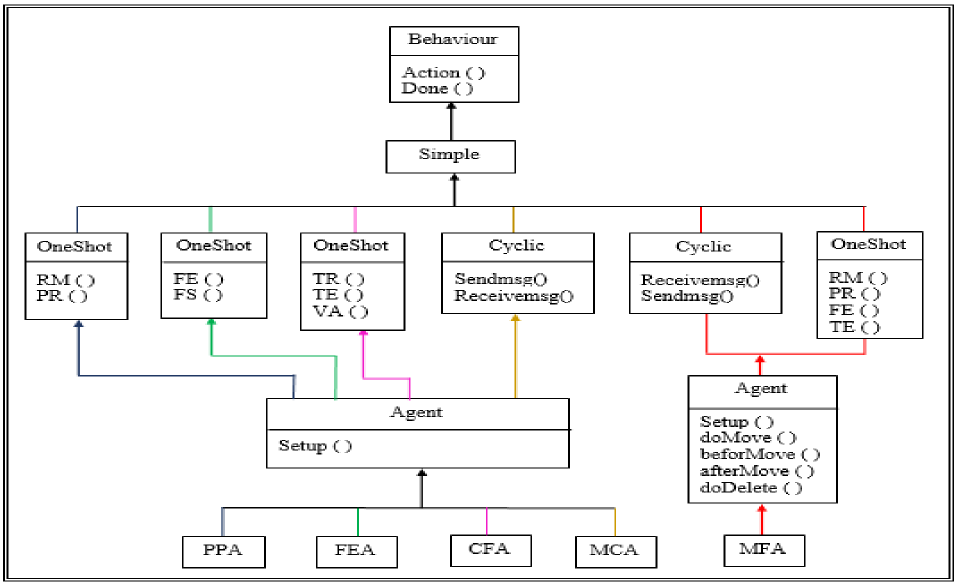


Figure (3): Low Level View

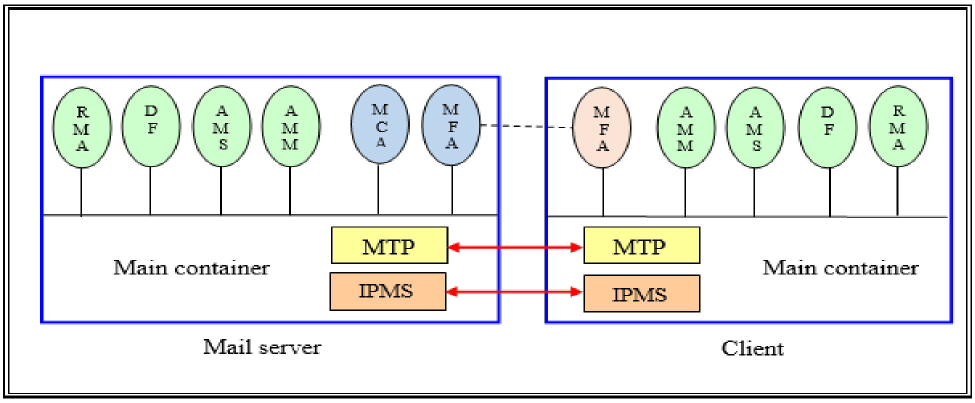


Figure (4): High Level View

4.2 Off-Line Classification: Experiments on CSDMC2010 Dataset

Working in the off-line mode, MASF system does classification on the bases of using CSDMC2010 spam e-mails dataset. Performance evaluation metrics used are: accuracy, recall, precision and F- measure are shown in Table(2).

Table (2): Performance of CSDMC2010 spam e-mails dataset

Accuracy	Recall	Precision	F-measure
0960739	096158195	098041475	09709703

4.3 On-Line Classification: Experiments on Real Data Samples

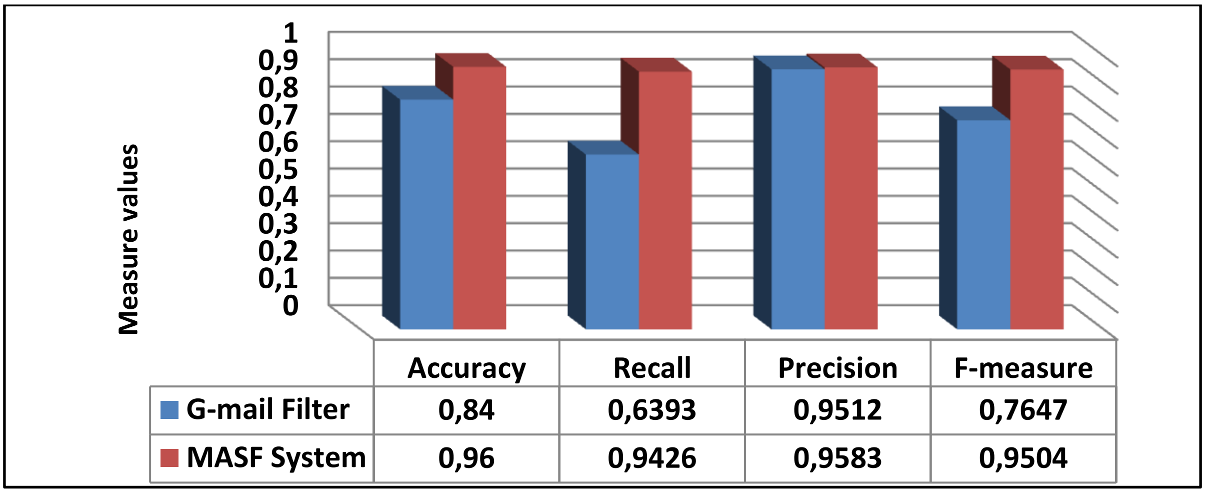
Working in the on-line mode, MASF system does classification on the bases of using real data samples: a total of 300 raw e-mail messages are collected manually for the on-line testing process, this data samples are composed of 178 non-spam e-mail message and 122 spam e-mail messages. Table (3) and Table (4) exhibit the confusion matrix gained upon on-line testing by Gmail-only filter and the MASF system respectively. Further, Figure (5) exhibits a comparative performance of both Gmail-only filter and MASF system on the foresaid real samples in terms of accuracy, recall, precision and F-measure.

Table (3): Confusion Matrix of Gmail Filter

Confusion Matrix		Predicted	
		Spam	Non-spam
Actual	Spam	78	44
	Non-spam	4	174

Table (4): Confusion Matrix of MASF System

Confusion Matrix		Predicted	
		Spam	Non-spam
Actual	Spam	115	7
	Non-spam	5	173



	Accuracy	Recall	Precision	F-measure
🟦 G-mail Filter	0,84	0,6393	0,9512	0,7647
🟥 MASF System	0,96	0,9426	0,9583	0,9504

Figure (5): Performance Gmail and MASF System

5.Conclusion

This work presented the design and implementation of spam filter system based on mobile agent technology with SVM-filter. The mobile agent is distributed between server and clients to identify e-mail messages as being spam or non-spam. The proposed system has been tested through two modes: “off-line” and “on-line”, in both modes, two datasets are used for implementation of proposed system: in “off-line” mode, a total of 4327 e-mail messages from CSDM2010 spam dataset are considered; 70% for training and 30% for testing, results showed that, SVM off-line classification gained accuracy rate of 0.9607%. For “on-line” mode, a total of 300 e-mail messages collected manually are considered for testing, and the system accuracy rate is 0.96% which remarkably outperform Gmail filter precisely in lowering the FP to about 2.3% of the total number of on-line samples. Java Agent Development Environment (JADE) is used to implement the proposed Multi-Agent based Spam Filtering (MASF) System.

It is realized that adopting agent-based design is rewarded to design and implement MASF system as it lends itself to wards autonomy, and that mobility greatly assisted distant filtering at client side; in that it preserved the accuracy achieved off-line while providing the needful flexibly to work in the dynamic environment.

References

Abdul, A. B. S. (2011). SVM Classification and Analisis of Margin Distance on Microarray Data.M.Sc Thesis, University of Akron.

Agarwal, D. K., & Kumar, R. (2016). Spam Filtering using SVM with different Kernel Functions. International Journal of Computer Applications, 136(5), 16–23.

Balakrishnan, S., &Shunmuganathan, K. L. (2015). An Agent Based Collaborative Spam Filtering Assistance Using JADE. International Journal of Applied Engineering Research, 10(21), 42476–42479.

Diale, M., Walt, C. Van Der, Celik, T., & Modupe, A. (2016).Feature Selection and Support Vector Machine Hyper-Parameter Optimisation for Spam Detection. IEEE Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech), 1–7.

Flaih, L. R., &Shukur, M. H. (2018).Software Agent for E-mail Spam Filtering. International Journal of Applied Engineering Research, 13(10), 7701–7703.

Hussain, Z. F. (2014). Design and Implementation of Multi- Agent System to Perform Electronic Examination System.M.Sc Thesis, College of Computer Science, University of Technology.

Jafar, A., &Edalatirad, M. (2015). Improving anti-spam filtering, based on Naive Bayesian and neural networks in multi-agent filters. Journal of Applied Environmental and Biological Sciences, 5, 381–386.

Kolluru, P. K. (2013). SVM based Dimensionality Reduction and Classification of Hyperspectral Data.

Liten, L. (2012). Communication and migration of an embeddable mobile agent platform supporting runtime code mobility. International Journal of Advanced Computer Science and Applications (IJACSA), 3(1), 49–54.

Mohammed, M. A., Shamini, S., Mostafa, S. A., & Ghani, M. K. A. (2018).Implementing an Agent-based Multi-Natural Language Anti-Spam Model. IEEE International Symposium on Agent, Multi-Agent Systems and Robotics (ISAMSR), 1–5.

Shajideen, N. M. (2018). Spam Filtering: A Comparison Between Different Machine Learning Classifiers. IEEE Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), 1919–1922.

Singh, V. K., & Bhardwaj, S. (2018). Spam mail detection using classification techniques and global training set. Springer Nature Singapore Pte Ltd., 623–632. https://doi.org/10.1007/978-981-10-7245-1_61

Teli, S., &Biradar, S. (2014).Effective Spam Detection Method for Email. International Conference on Advances in Engineering & Technology (ICAET), 68–72. Retrieved from http://www.iosrjournals.org/iosr-jce/papers/ICAET-2014/volume-4/12.pdf

Vyroubal, V., &Kušek, M. (2013).Task Migration of JADE Agents on Android Platform. IEEE 12th International Conference on Telecommunications, (June 2013), 123–130.