World Engineering & Applied Sciences Journal 7 (3): 155-162, 2016
ISSN 2079-2204

DOI: 10.5829/idosi.weasj.2016.7.3.22668

Collaborative Pattern-Based Filtering Algorithm for Botnet Detection

[1]P. Panimalar and [2]K. Rameshkumar

[1]Research Scholar, Research and Development Centre, Bharathiar University, Coimbatore-641046, India
[2]Research SupervisorResearch and Development Centre, Bharathiar University, Coimbatore-641046, India

Abstract: Botnet is a malicious program which is actively engaged internet crimes. It is considered as one of the major Internet threats in recent years. Signature and behaviour based techniques are two major patterns used in the botnet detection. The behaviour based technique helps to find the unknown variants of bots. In this paper, we propose a Collaborative Pattern based Filtering (CPF) algorithm which is a behaviour based approach to detect bots in association with Case Based Reasoning (CBR) and fuzzy pattern recognition techniques. Network traces are used as a pivotal element to inspect bot relevant domain names and IP addresses. The proposed approach reduces the search time, enhances the prediction accuracy of 96% and it is also observed that the increase of knowledge repository has found significant relationship with reduction of search time.

Key words: Botnet detection · Collaborative filtering · Case-based reasoning · Fuzzy pattern recognition

INTRODUCTION

Botnet is a collection of compromised computers which are organized to do malicious activities. The word botnet is derived from the combination of the words robot and network. Botnet is considered as one of the major security threats and it does not affect the regular usage of negotiated computers instead it runs in hidden mode and it processes variety of malicious activities such as stealing personal data, credit card details, sending SPAMs, spyware and adware. The botnet contains bot herder, Command and Control (C&C) server, bot and target host and botnet program works on client/server architecture. It is generally using the Internet Relay Chat (IRC) channel for communication [1]. The main process of botnet is to commence Distributed Denial of Service (DDoS) attacks on the target host, which are simultaneously executed from multiple zombies (infected machines) under the command and control of a bot herder. The objective of the attack is to slow down the target host, network or application. Thus, target host can't respond to genuine requests. Similarly, attack may also install malware to steal valuable information from the zombies.

Due to the rapid growth of botnet and its threatening, many researchers have been involved in the investigation of botnet architecture [2], detection [3]and prevention methodologies. This research is focus on extending the feature of botnet detection technique. The previous studies have states that the botnet detection system is majorly classified into two types such as signature based detection and behaviour based detection. In an ideal scenario, signature based detection system is more accurate than behaviour based detection whereas it is failed to predict the unknown threats. Therefore, well trained behaviour based detection is not only improves the prediction quality, it also provides lightweight solution for botnet detection.

In this paper, a collaborative pattern-based filtering (CPF) algorithm is proposed to detect malicious domain names and IP addresses used by botnets. The contribution of the paper is presented in two sections. First section describes a framework which combines case based reasoning (CBR) and fuzzy pattern recognition for effective botnet detection. Second section describes the empirical evaluation of proposed behaviour based botnet detection technique performance.

Corresponding Author: P. Panimalar, Research Scholar, Research and Development Centre, Bharathiar University, Coimbatore-641046, India.

Literature Review: Botnet is a malicious program intensively sent to many people as SPAM mail, spyware and adware. Computer or network with weak security features may allow exploiting the vulnerable software in the local machine. Such a compromised computer is known as bot, which does not affect the regular usage. Bot herders are frequently communicating with bots through C&C servers to lookup the valuable information in the target hosts. Network traffic reduction is an important step to improve the overall system performance. Among the list of traffic reduction methods, eliminating port scan activities does not affect the prediction quality; thus, port scan activities are filtered. Understanding the bot behaviour is an essential step to identify the unknown variants.

Wang *et al*., [4] proposed a behaviour based botnet detection system using fuzzy pattern recognition techniques. Bot relevant domain names and IP addresses are identified through inspecting the network traces. FPRF has considered four behaviours such as (1) generate failed DNS queries (2) have similar query intervals (3) generate failed network connections and (4) have similar payload size for network connections. It follows membership functions to predict the probability of being malicious or benign IP address. The evaluation result states that FPRF algorithm can reduce more than 70% input raw packet traces and achieves a high detection rate 95%. It is suggested as resource efficient and can identify inactive botnets.

Burke.R [5] proposed collaborative filtering systems make recommendations based on the accumulation of ratings by many users. The process has a case-based reasoning flavour: recommendations are generated by looking at the behaviour of other users who are considered similar. However, the features associated with a user are semantically weak compared with those used by CBR systems. This research examines multi-dimensional or semantic ratings in which a system gets information about the reason behind a preference. Experiments show that metrics in which the semantic meaning of each rating is taken into account have markedly superior performance than simpler techniques.

Garcia, Sebastian *et al*., [6] were examined botnets behaviours detection using network synchronisation. They have used inherent characteristics, like synchronism and network load combined with a detailed analysis of error rates which is not relying in any specific botnet technology or protocol. The classification approach sought to detect synchronic behavioural patterns in network traffic flows and clustered them based on botnets characteristics. Different botnet and normal captures were taken and a time slice approach was used to successfully separate them. The final outcome of this approach states that botnets and normal computers traffic can be accurately detected.

Zia, Syed Saood *et al*., [7] examined the case based reasoning (CBR), which ensures the accuracy of decisions during diagnosis and treatment phases of patient care. This paper explores case retrieval phase of the CBR technique which is applied on breast cancer dataset UCI Machine Learning Repository.

Problem Statement: Botnet is the most vulnerable threat in the recent era of internet based developments and its statistics shows that number of botnet is increasing [8]. Hackers were using many techniques to attack the network. Thus, understanding the common behaviour of botnet is essential to predict the malicious movements. Recent proposal by Wang *et al*., [4] insisted the fuzzy pattern-based filtering algorithm for botnet detection. It is truly an embracing technique to predict the real-time botnets. The FPRF algorithm is mainly trying to identify the domain names and IP addresses used by bot C&C servers. To achieve the result, FPRF has splits the processes into three phases known as traffic reduction, feature extraction and fuzzy pattern recognition. Traffic reduction is the initial phase which uses intrinsic filtering method to eliminate the packets relevant to botnet detection. Feature extraction is the second phase which extracts the observable features from the input traces such as query interval times andpayload size. In the third phase, fuzzy pattern recognition technique is applied to predict the givendomain name or IP address is malicious using set ofbotnet behaviours considered in his paper.

The preliminary analysis of FPRF algorithm exhibits (Figure 1 illustrates the FPRF architecture) as each query was end up with the fuzzy patternrecognition phase in order to identify the given instruction is malicious or not. Even though a simple arithmetic calculation was used in the membership function, the fate is to process all the queries given for evaluation. Further, feature extraction phase consumes considerable amount of time to observe the features from the input traces. Due to the nature of probability function, the prediction ratio of DNS queries or IP addresses may differ each time. Hence, the main objective of this research is to provide a new solutionwith

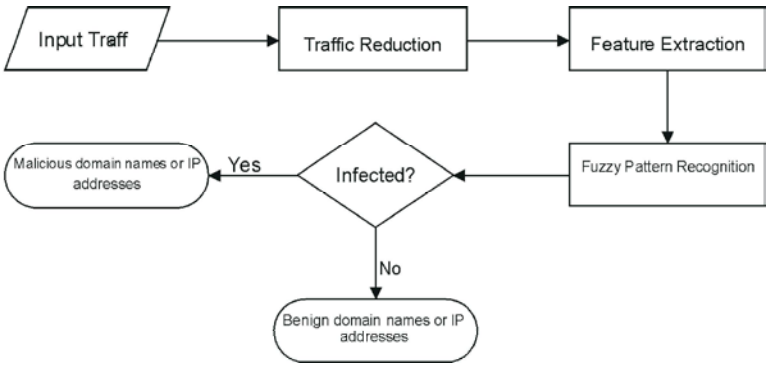

Fig. 1: Fuzzy pattern recognition based filtering (FPRF) algorithm.

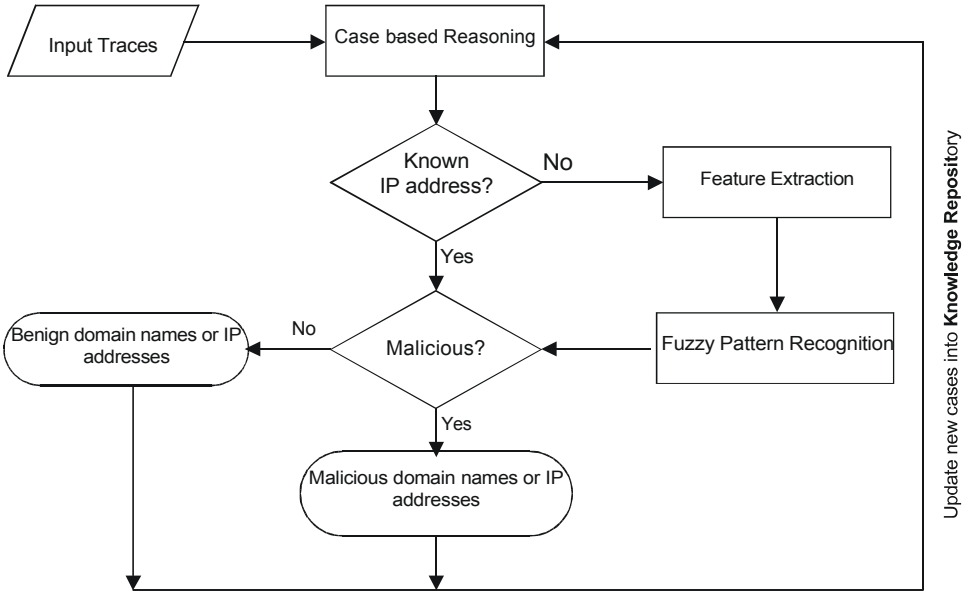

Fig. 2: Collaborative pattern-based filtering (CPF) algorithm for botnet detection.

reduced process time and enhance the quality of prediction accuracy. Subsequent section describes our proposed method for botnet detection based on the selected bot behaviours.

Collaborative Pattern-Based Filtering (CPF) Algorithm: The main intention of our proposal is to optimize the time without compromising the quality of output.The proposed system contains three phases known as knowledge repository (case-based reasoning), feature extraction and fuzzy pattern recognition. In this paper, trace files are given as input for verifying and validating the complete proposal. The input tracesimmediately verify the availability of domain name or IP address in the knowledge repository; if found itclassify the status. Whenunknown domain names or IP addresses are encountered, the query redirect to feature extraction phase, which extracts the behaviours based on the membership function defined [4]. Finally, unknown queries are sending to fuzzy pattern recognition phase to classify the query status of being malicious. Figure 2 illustrates the proposed architecture of botnet detection.

Bot behaviour is an eminent characteristic which enables to predict the bots correctly. In this paper four different behaviours are considered for evaluation [4].

- Generate Failed Domain Name System (DNS) Queries: A bot generally has a built-in domain name list to all possible C&C servers. To prevent from being detected the C&C servers often changed its mode to off-line or shutdown immediately. During this time bot cannot respond the DNS query requested by C&C server, hence it generates a failed DNS response.

- Query Intervals: If a DNS query response failed, a bot may lookup either the samedomain name again or the next domain name available in the built-in domain name list. Bot trying to contact C&C server at frequent interval, hence considering query interval helps to identify the bot.
- Generate Failed Network Flows: Similar to domain name list bot also uses IP address list maintained by them when a bot tries to contact unreachable C&C server. Otherwise, IP addresses obtained from DNS server cannot contact by the computer. In both scenarios, failed network flow is generated.
- Similar Payload Size for different network flows: Bots generally has multiple payloads including SYN, UDP, GET and DNS. But, when a bot reached the C&C server successfully, it tries to download the commands sent by bot herders. This command remains unchanged for specific period due to the uncertainty of bot. As stated by Wang *et al*., [4], the payload size for a TCP and an UDP network flow is countered in different ways.

The case based reasoning (CBR) method generally using the similarity function to validate the closeness of new case. In our proposal, instead of predicting through similarity and distance function we suggested to use the fuzzy pattern recognition technique[5-8].

Case Base Reasoning (CBR) Phase: The case based reasoning is one of most successful applied artificial intelligence technologies of recent years. CBR has intuition that new problems are often similar to previously encountered problems and hence past solution may be useful for the current situation [9]. The foremost factor that ensures performance of CBR systems is a proficient way to retrieve cases from the case repository [10].

In general, the CBR system is termed as CBR cycle, which contains four phases namely retrieval, reuse, revise and retain. Retrieval phase is an initial step which inquires about previous experiences that are similar to the new case. This phase extracts most similar cases from the case repository. Reuse is the second phase which is responsible in suggesting a solution for the new case from the available solutions of the cases that were retrieved from the case repository. Revise is the third phase which happens when a distinct feature extracted for future use by experts or automatic. Retain is final phase of CBR cycle which tries to retain the new case for future usage. The fundamental process model shown in Figure 3 was identified by Aamodt *et al*. in 1994 [11].

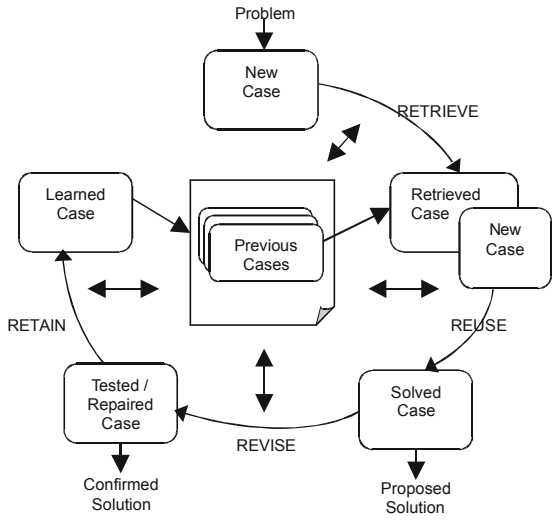

Fig. 3: CBR cycle [Aamodt, 1994].

The proposed collaborative framework operates with case-based reasoning and fuzzy pattern recognition techniques. In this proposal, CBR takes a major role in detecting bots with respect to the previous knowledge and when it attempts to predict the unknown threats it uses fuzzy pattern recognition which is described in section 4.3. Knowledgerepository is the central part of CBR, which maintains the case history of previous instances and it maintains two lists of case histories known as master list and probable list. The knowledge repository seems empty for a new launch, thus it ignite the featureextraction process. After observing the set of botbehaviours, it tries to evaluatewith fuzzy patternrecognition to classify the status of input. Such outputs are generally stored in the probable list. Due to the nature of probability used in the membership function the outputratio may vary each time. In order to unify the variance, maturity threshold m insists the point to call optimize function. Similarly, precision threshold p is defined to declare the domain name or IP address is malicious or benign. The optimization function is defined as follows; x is a set of probable cases with reference to the specific domain name or IP address, therefore $\mu = (x,m,p)$.

$$f(\mu) = \begin{cases} \frac{\Sigma x}{m}, & \frac{\Sigma x}{m} \geq p, \\ \frac{\max(x)}{\Sigma x}, & otherwise. \end{cases} \quad (1)$$

If the optimized value of mean is greater than precision threshold then it is strongly declared to be the same state as earlier, either malicious or benign. Otherwise, decision can be made using proportional value

using max function in the probable list. Once optimized value is generated, the corresponding case histories in the probable list are being removed and most of this process will be done at revise phase.

The retrieval is an important element in CBR, which extracts the similar case histories from the knowledge repository and its best match is predicted by reuse phase using the following function.

$$R = max(\alpha, p) \quad (2)$$

Where, α is a set of related domain names or IP addresses and p is the precision threshold stored in the master list. The retrieval phase primarily concentrates on master list, if not found then it lookup the probable list. Retain is the final phase which stores new case histories until it reaches the limit; once reached, it removes the least recently used instance and allow a space for new case known as obsolete function.

The entire case based reasoning can be presented in form of pseudo code as follows:

Pseudo Code of CBR:
Input: Master List Ml, Probable List Pl, Feature Vector Fv
Output:Boolean
Start
  Read Master List Ml.
    For each entry Mi from Ml
      If $\int_{i=1}^{size(Ml)} \forall(Fi(mi) \equiv Fi(Fv))$ then

  Boolean=Perform Fuzzy Pattern Recognition(Fv)
  If true then
    Compute Precision value Pv.
    Pv = $\int_{i=1}^{size(Ml)} \frac{\sum Ml(i)==Fv}{size(Ml)}$

    If Pv>=Precision Threshold then
      If $\int_{i=1}^{size(Ml)} Ml \exists Pv$ then

        Add to malicious list.
        Ml = $\sum(Mi \in Ml) \cup Pv$

      Else
        Add to probability list Pl.
        Pl = $\sum(Pi \in Pl) \cup Pv$

      End
    End
Else
    Add to probability list Pl.
        Pl = $\sum(Pi \in Pl) \cup Pv$

End
Stop.

The above discussed case based reasoning algorithm performs the reasoning of packet being received with the help of fuzzy pattern recognition and feature extraction methods.

Feature Extraction Phase: This phase extract features from the DNS queries which are used to measure the quality of being malicious or normal. Two different input traces has been collected for evaluation namely DNS packet traces and network flow traces. Packet traces helps to identify the interval pattern of DNS queries, since bots usually follows fixed interval pattern and sometimes it may work on interleaved pattern. Similarly, the network flows trace helps to identify the relationship between DNS queries and network flows.

Pseudo code of Feature Extraction:
Input:Flow Trace Ft, Network Trace Nt, Packet P
Output: Feature Vector Fv
Start
  Extract packet features Pv = $\sum Features \in P$

  Read flow trace Ft.
  For each time window Ti from Tw
    Identify distinct flow pattern from same source.
    Fp = $\int_{i=1}^{size(Tw)} \sum Distinct(Fi) \in Ft$

    Identify distinct DNS pattern from same source.
    Dp = $\int_{i=1}^{size(Tw)} \sum Distinct(Ni) \in Nt$

  End
    Construct Feature Vector Fv.
    Fv = $\sum Features\ (P) \cup Fp \cup DP$

Stop

The feature extraction phase extracts the features of the packet being received and also extract the pattern from the two different traces being used. The extracted features are constructed as feature vector and has been used in case based reasoning phase.

Fuzzy Pattern Recognition Phase: Bots mostly preferred to work on dynamic basis to increase the complexity of being detected from the traditional way. Keen observation of behaviour provides more insights about the bot, thereby it can be resolved from the computer. Fuzzy pattern recognition technique is used to detect bots by packet level traces of DNS queries and network flows.

DNS Phase: Feature vector is defined for a domain name as x = (α, β, γ), where, α as fixed size set that contains n counters and each counter has an initial value of zero.

The network traces containing m DNS queries, the time intervals is estimated from the difference of two successive DNS queries which form a sequence $S = \{s_1, s_2, ..., s_{m-1}\}$. For each time interval s_j (1= j= m-1) in S, we calculate i as $[s_j]$ if and only if s_j is less than or equal to n and then increase the counter a_i by 1. Therefore, if s_j is greater than n, no counter is increased. β is the total number of DNS responses and γ is the number of failed DNS responses. In this phase, three membership functions are defined to identify the state of DNS query.

Inactive malicious DNS query: The inactive malicious domain name can be identified through the failed DNS responses. Since, compared to the normal DNS query, malicious domain name has more number of responses. The following membership function X_1 is used to calculate the probability of being an inactive malicious DNS query.

$$X_1(x) = 1 - \frac{\beta - \gamma}{\beta} \quad (3)$$

Malicious DNS query: A membership function is defined to predict the malicious DNS query based on its time intervals. Hence, it is assumed that if an identified domain name has similar time interval then it is declared as malicious. The following membership function X_2 is used to calculate the probability of being a malicious query.

$$X_2(x) = \begin{cases} \frac{max(\alpha)}{\Sigma \alpha}, & \Sigma \alpha \geq p \\ 0, & otherwise. \end{cases} \quad (4)$$

Normal DNS Query: The probability of being normal query is calculated using X_3 as follows;

$$X_3(x) = 1 - max\{X_1(x), X_2(x)\} \quad (5)$$

Network Flow Phase: The feature vector is defined for network flow as x = (α, β, γ) where, α as fixed size set that contains n counters and its initial value is zero. β is the total number of network requests. If the maximum payload size is less than b bytes, γ is defined as fixed size that contains b + 1 counters and each counter has initial value of zero. A network trace containing t networkflows and its payload size of each network flow is extracted and form a sequence $P = \{p_1, p_2 ... p_t\}$. For each payload size p_j (1 = j = t) in P, we set i to p_j and then increase the counter r_i by 1 if i = b. The following three memberships function are used in this phase.

Inactive Malicious IP Address: A symptom of an IP address receives many requests but does not respond is majorly denotes as inactive malicious IP address. In the following equation ó is a threshold. The membership function X_1 defined as follows;

$$X_1(x) = \begin{cases} 1, & \Sigma \alpha = 0 \text{ and } \beta \geq \sigma, \\ 0, & otherwise. \end{cases} \quad (6)$$

Malicious IP Address: The well-known malicious IP address has been identified using its similar payload sizes. Hence the membership function X_2 is defined as follows;

$$X_2(x) = \begin{cases} \frac{max(\gamma)}{\beta - r_0}, & \beta - r_0 \geq \rho \\ 0, & otherwise. \end{cases} \quad (7)$$

Normal IP Address: The probability of being normal IP address is defined using X_3 as follows;

$$X_3(x) = 1 - max\{X_1(x), X_2(x)\} \quad (8)$$

The CPF proposal has not only reduces the time taken to predict the botnet prediction, in addition it is maintaining the local repository of blacklist and whitelist of domain names or IP addresses for future use. In order to avoid huge volume of unused or merely less used domain names or IP addresses, we use an auto obsolete method to limit the knowledge base. The subsequent section explains the performance results.

Pseudo Code of Fuzzy Pattern Recognition Phase
Input: Feature Vector Fv, Malicious List Ml, Probability List Pl
Output: Boolean
Start
  Read Feature Vector Fv.
  Compute Probability ofNomal DNS query x1.
  Compute Probability of Inactive DNS query x2.
  Compute Probability of Malicious DNS query x3.
  Compute Probability of Normal IP Address x4.
  Compute Probability of Inactive IP Address x5.
  Compute Probability of Malicious IP Address x6.
  Compute Normal probability Np = X1×X4
  Compute Inactive probability Ip = X2×X5
  Compute Malicious probability Mp = X3×X6
  For each case
  If probability <> Fuzzy Range then
    Return true
  Else
    Return false.
  End
  End
  Stop.

The presented pseudo code computes probability of normal, inactive and malicious DNS query as well as on IP address using the network and flow trace available. Based on computed probability a single case is being selected and returned as result.

RESULTS AND DISCUSSIONS

An attempt has been taken to collect real botnet traces with the help of selected malicious binaries. The input traces are given as flat file which has two splits namely requests & flows and its outputs are stored in a MySQL database. The request set has source, destination IP addresses and its associated network requests. The flows set contains detailed information regarding the IP addresses, port, time, file size, topology and its status. The original dataset size is approximately 1.2 GB and we have written a separate program in PHP language to split the dataset into required number of files in random basis. For our convenience, we have extracted four subsets from the main dataset known as scenario(s) in this paper. The test scenario of maturity threshold value is set as 5 and its precision threshold value is 50%.

Figure 4 illustrates the time taken to process query by FPRF algorithm and CPF algorithm in four scenarios considered for evaluation. The result states that S1 time seems very close to one another and in this scenario CPF algorithm consumes more time compared to FPRF. There is no case history exist in this situation, therefore it consumes extra time to writenew cases. The S2, S3 and S4 scenario illustrates a drastic difference of time taken to process a query. Time taken to process query by FPRF algorithm as follows; S1=27sec, S2=21sec, S3=34sec, S4=26sec and time taken to process query by CPF algorithm are S1=29sec, S2=17sec, S3=16sec, S4=14sec.

In this paper, the accuracy is observed using the total number of malicious domains or IP addresses divided by the number of domain names or IP addresses using FPRF and CPF algorithms.

Figure 5 exhibits the botnet prediction accuracy delivered by FPRF and CPF algorithms among the four scenarios considered in our empirical evaluation. It is observed from the result that CPFalgorithm provides slightly better accuracy compared to FPRF algorithm. This investigation states that FPRF algorithm is purely predicts using membership function which depends on the probability function. Whereas the CPF algorithm holds the existing prediction variants of same domain or IP address either inactive malicious query or malicious query. Therefore it reduces only the misclassification errors in the probability function, which is a great asset of CPF prediction accuracy.

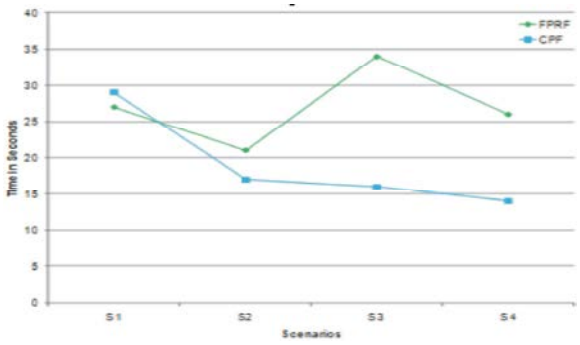

Fig. 4: Time taken to process a query.

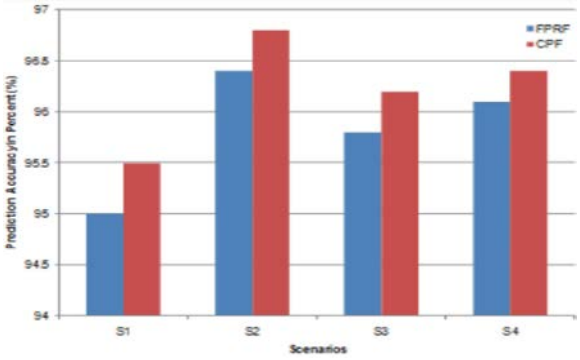

Fig. 5: Botnet prediction accuracy.

Table 1: T-test analysis – Prediction accuracy.

Algorithms	Mean	SD	t-value	p-value
FPRF	95.84	0.61	6.069	0.009*
CPF	96.21	0.49		

*Significant at 5% level

Prediction result is further analysed using statistical function t-test to test the significant difference on the prediction accuracy. It is observed from the Table 1 that FPRF algorithm has prediction accuracy of 95.84±0.61 and CPF algorithm has prediction accuracy of 96.21±0.49. The calculated t-value is 6.069 and its p-value is 0.009, which is less than the level of significance 0.05. Hence, the result confirms the significant difference between two algorithms and the key finding states that CPF algorithm outperforms.

CONCLUSION

In this paper, we propose Collaboration Pattern-based Filtering (CPF) algorithm which is an extension of existing Fuzzy Pattern Recognition based Filtering (FPRF) algorithm. The CPF algorithm uses Case Based Reasoning (CBR) technique to prepare and maintain the knowledge base observed from the series of previous executions.

Therefore it avoids predicting the same domain names or IP addresses which are well known. The empirical evaluation mainly targets on reducing the time taken to predict the bots and improve the quality of prediction. The proposed CPF algorithm outperforms in time taken to predict the bots. Similarly it is also maintain the significant difference and improvement in the botnet prediction accuracy.The proposed solution is limited to predict bots based on the local repository. The CPF algorithm uses only limited resources and it is capable of predicting inactive, new bots. This method is highly suggested for cost effective network security and intrusion detection system.

REFERENCES

1. Feily Maryam and Alireza Shahrestani, 2009. A Survey of Botnet and Botnet Detection, in Proceedings of the Third International Conference of IEEEon Emerging Security Information Systems and Technologies, pp: 268-273.
2. Wang Ping, Sherri Sparks and Cliff C. Zou, 2010. An Advanced Hybrid Peer-to-Peer Botnet, IEEE Transactions on Dependable and Secure Computing, 7(2): 113-127.
3. Gu, G., J. Zhang and W. Lee, 2008. BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic,in Proceedings of the 15th Annual Network and Distributed System Security Symposium.
4. Kuochen Wang, 2011. A Fuzzy Pattern-based Filtering Algorithm for Botnet Detection, Computer Networks, 55(15): 3275-3286.
5. Burke, R., 2007. Hybrid Web Recommender Systems, The Adaptive Web, 4321 (The adaptive web), pp: 377-408.
6. Sebastián García., 2011. Botnet Behavior Detection using Network Synchronism, Privacy, Intrusion Detection and Response: Technologies for Protecting Networks: Technologies for Protecting Networks, pp: 122-144.
7. Zia, Syed Saood, 2014. Case Retrieval Phase of Case-Based Reasoning Technique for Medical Diagnosis, World Applied Sciences Journal, 32(3): 451-458.
8. McCarty, B., 2003. Botnets: Big and Bigger, IEEE Security and Privacy, 1(4): 87-90.
9. Bjork, B.C., 2004. Open Access to Scientific Publications – An Analysis of the Barriers to Change?,Information Research, 9(2).
10. Lopez De Mantaras, 2005. Retrieval, reuse, revision and retention in case-based reasoning, TheKnowledge Engineering Review, 20(3): 215-240.
11. Aamodt A. and E. Plaza, 1994. Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches, AI communications, 7(1): 39-59.