IJASRM

International Journal of Advanced Scientific Research and Management, Volume 3 Issue 11, Nov 2018

www.ijasrm.com

ISSN 2455-6378

Fuzzified Multi Layered Deep Neural Network (FMLDNN) for Anomaly Detection in DNS Query Logs

Ms.S.S.Suganya[1], Dr.V.Kathiresan[2]

[1] Department of Computer Science, Dr.SNSRajalakshmi college of Arts & Science, Coimbatore, Tamil Nadu, India.

[2] Department of Computer Applications(PG), Dr.SNSRajalakshmi college of Arts & Science, Coimbatore, Tamil Nadu, India

Abstract

Web mining includes two major tasks namely predictive mining and descriptive mining. Anomaly detection in DNS query logs stands a big research issue in the field of web mining. From the literatures, it is observed that several conventional web mining algorithm does not significantly obtained better accuracy more than ~80%. In this research work, fuzzy based multi layered deep neural network is employed for performing the anomaly detection in DNS query logs in order to perform the classification task. FMLDNN outperforms than that of the compared algorithms and attains maximum of 95.88% classification accuracy with reduced time consumption.

Keywords: DNS, data mining, query logs, neural network, deep learning, fuzzy logic, classification accuracy, time taken.

1. Introduction

DNS (Domain Name System) is the protocol which is widely used in several varieties of internet and its applications that includes the usage in sending and receiving emails, transfer of files / data using File Transfer Protocol (FTP) and configuring and accessing the internetwork / intranetworking with the help of Simple Network Management Protocol (SNMP). Initially, the Uniform Resource Locator (URL) denotes the area of a web resource has a domain name string. Accurately when a node (it is possible that it might be a PC or cell phone) tries to make utilization of the internet resource, the hub needs to decide the right IP address of the relating domain name. Furthermore, in mail server condition, the email id comprises of client name and a domain name, which is utilized by the mail server. Consequently, DNS inquiries are profoundly reliant to hubs conduct that incorporates web perusing or utilizing messages. As the limit of internet activity is snowballing step by step, it is exceptionally extreme errand to screen singular hubs that are associated through Internet. DNS is useful for continue following hub irregularities in a network organization. There are whimsical strategies and procedures being utilized to monitor DNS movement. Jung et al. suggested that DNS movement be accessible to watch inconsistencies [1]. Conversely, the DNS servers additionally get influenced by a few assaults, for example, a DDoS (Distributed Denial of Service) assault and a reserve harming assault. DNS is a protocol on UDP and furthermore TCP in which the source IP addresses can be spoofed by an aggressor with the goal that the assailant can abuse DNS. For instance, a DNS server is conceivably a reflector of a DRDoS (Distributed Reflected DoS) assault questioned spoofed parcels with the focused on have.

Query logs contain only the queries that DNS resolvers forward to certain route with routeid. When a DNS resolver has already cached the response to a query (such as the IP address for a load balancer for example.com), the resolver will continue to return the cached response without forwarding the query to the certain routeid until the TTL for the corresponding record expires. Depending on how many DNS queries are submitted for a domain name (example.com) or subdomain name (www.example.com), which resolvers your users are using, and the TTL for the record, query logs might contain information about only one query out of every several thousand queries that are submitted to DNS resolvers. This research work aims to present fuzzy based multi layered deep neural network. Deep neural networks use sophisticated mathematical modeling to process data in complex ways. The phrase “deep learning” is also used to describe these deep neural networks,

as deep learning represents a specific form of machine learning where technologies using aspects of artificial intelligence seek to classify and order information in ways that go beyond simple input/output protocols.

2. Related Works

Mama et al. [2] proposed a work that plans to appraise DNS question attributes in view of DNS store exercises. The information have been gotten with the assistance of dynamic examining on an extensive scale at insignificant management cost and limited protection concerns. At initial, a novel arrangement that incorporates the restoration hypothesis based DNS caching plan and the hyper-exponential appropriation display. After that they played out a huge scale true DNS follow estimation, and showed that their answer amazingly enhances the estimation exactness lastly they connected their answer for evaluate the malware-tainted host populace in remote management networks.

Q. Lai et al. [3] endeavored to tour the practices of DNS query by exhuming DNS logs from three essential DNS servers in a heavy college grounds network. Their dataset has two sections, namely DNS question logs and messages got or send by DNS servers. At to begin with, finished investigating the DNS inquiry logs, the creators expressed that, they could grasp the general propensity of clients' perusing. Keeping in mind the end goal to manage immense DNS dataset, the creators presented a calculation and named it as DNS Reduce that is utilized to perceive top 10 customer IP addresses and best 10 goal domain names ably. At that point with the assistance of messages got or send by DNS servers the creators decided the DNS servers′ practices and asserted that they can precisely recognize domain names with dynamic IP addresses. The creators additionally gave different and particular representation strategies for displaying their analysis.

Y. Jin et al. [4] proposed a strong creation by making utilization of DNS for doing the goal. In their work, IDs of Internet of Things (IoT) gadgets are adapted by DNS server and the observing and control are led by the coordinated effort of DNS name determination, DNS dynamic refresh and DNS zone transfer. Assessing security and security insurance, the status and control summon for IoT gadgets portrayed in the relating DNS TXT records are scrambled and TSIG (Transaction SIGnatures) are utilized for confirmation to restrict the customer machines and are likewise permitted to screen and control the IoT gadgets.

B.S. Fagin et al. [5] proposed a compelling and pertinent DNS burrow detection instrument. Their model system is conveyed at the Recursive DNS for burrow recognizable proof. The creators made utilization of four sorts of highlights namely time-interim highlights, ask for parcel measure highlights, record compose highlights and subdomain entropy highlights. Their proposed system has been assessed the execution with Support Vector Machine, Decision Tree and Logistical Regression. In their closing comments the creators expressed that SVM will perform superior to anything that of other machine learning calculations.

N.F. Haung et al. [6] exhibited an application recognizable proof system that consolidates a QoS management system in a software characterized network (SDN). Their manuscript depicted the strategy to procure ground truth (name) of the spill out of four standard working systems (OS), and the technique to group stream in view of managed machine learning and DNS reactions. In their execution assessment, normal F-measure of all applications achieved 93.48%. Their testing informational collection contained 294 applications, given that every stage adaptation or execution file of an application was one application. The testing informational index included Skype, Facebook, and other prevalent applications.

R. Romero-Gomez et al. [7] proposed a client - focused approach, beginning with outlining an open source threat analysis console for DNS-based network threat analysis grounded in both a comprehension of investigators' needs and undertakings and security representation best practices. Their proposed open source threat analysis console, called THACO (THreat Analysis COnsole), use open DNS datasets, domain WHOIS records, and both open malware and domain boycotts. THACO likewise utilizes an outwardly versatile perception system, a multi-gathering, zoomabletreemap, to adjust to DNS-based network threat analysis needs. At that point, the creators led a client think about with 7 in-situ and 31 online IT security professionals.

M. S. Manggalanny, K. Ramli [8] proposed an action of Advanced Persistent Threat (APT). New modus and systems were being produced quickly and conquered the push to recognize it. Their outline proposed another approach through a blend of a few fruitful detection strategies in view of DNS movement analysis by and large, to address late APT difficulties. Their preparatory investigation indicates promising and better exactness of APT acknowledgment and speedier reaction. From the above writings it is obvious that exclusive a not very many written works addressed

the anomaly detection utilizing DNS server sign as of late. Subsequently there is a wide extension for look into around there.

AlbrechtBuehler used motion to visualize trends among text–theme relationships and allowed user interaction of the temporal controls and theme relations [9]. Brandes et al. used animation to illustrate the dynamics of international political and military conflicts [10]. In Pieter's research, a visual analytics approach was used on a large set of DNS packet captures to gain insight into ways that authoritative name servers were abused for denial of service attacks [11]. Several tools were developed to identify patterns in DNS queries and responses. Yu presented a visualization analysis tool for detecting, analyzing and responding to the Distributed Denial of Service attack termed the Domain Name Service (DNS) amplification attack [12]. In Born's study both quantitative analysis and visual aids were provided that allowed the user to make determinations about the legitimacy of the DNS traffic [13].

Jung et al. proposed the Localized Big Data Analysis (LBDA) with which data mining techniques were applied to the DNS log [14]. Ruan et al. proposed a novel periodic trend mining method as well as a periodic trend pattern based traffic prediction method [15]. David Dagon described a tutorial on large-scale DNS data analysis, the tutorial was a pragmatic course on data collection, analysis and techniques [16]. Callahan et al. made an initial empirical understanding of a broad range of characteristics of modern DNS behavior, such as TTLs, equivalent answers and proximity [17].

3. Fuzzified Multi Layered Deep Neural Network (FMLDNN)

3.1. The model of layered innate-encoder (LIE)

An Innate-Encoder (IE) is a neural network with only one hidden layer and with the same number of nodes in the input and output layers. Given a set of training records $\{x^{(1)}, x^{(2)}, x^{(3)}, ...\}$, where $x^{(i)} \in R^d$, an IE first encodes an input $x^{(i)}$ to a hidden depiction $y(x^{(i)})$ computed by Eq. (1), and then decodes depiction $y(x^{(i)})$ back into a re-enactment $z(x^{(i)})$ computed by Eq. (2).

$$y(x) = f(C_1 \cdot x + b_1) \quad \dots (1)$$

$$z(x) = g(C_2 \cdot y(x) + b_2) \quad \dots (2)$$

where C_1 is a credence milieu, b_1 is an encoding bias vector, C_2 is a decoding milieu, and b_2 is a decoding bias vector. $f(x)$ and $g(x)$ are the activation functions. The innate-encoder can be trained using the conventional back propagation algorithm. Training of innate-encoder takes each input records itself as its label. The target is to minimize the re-enactment fault $F(X, Z)$ defined by Eq. (3).

$$F(X, Z) = \frac{1}{2}\sum_{i=1}^{N} \left\| x^i - z^i \right\|^2 \quad \dots (3)$$

where X is the set of N input records $\{x^{(1)}, x^{(2)}, ..., x^{(N)}\}$, and Z is the set of corresponding reconstructed output $\{z^{(1)}, z^{(2)}, ..., z^{(N)}\}$. And the notation $\| \cdot \|$ represents the 2-norm of a vector.

A LIE model is created by ordering innate-encoder to form a deep neural network. Considering a LIE with three layers and given a set of input records, the training of a LIE is straightforward. The first layer is trained as an innate-encoder, with the training set as inputs such that C_1^1 and C_2^1 are obtained. Then for each of others innate-encoder, the encoding of the first IE is taken as the input to train the next one such that C_1^i and C_2^i are obtained. In this way, the training of multiple innate-encoders is completed.

3.2. Back-propagation mechanism

Improvement is a approach that is commonly found in deep learning. It can be used to greatly improve the performance of layered innate-encoder. From a high level standpoint, improvement treats all layers of LIE as a single model, so that in one iteration, we are improving upon all the credence. As the back-propagation algorithm can be extended to apply for an arbitrary number of layers, we can actually use this algorithm on layered innate-encoder of arbitrary depth. In this work, to acclimatize the bonds credence in order to obtain minimal difference between the network output and the desired output, we used the back-propagation algorithm. BP algorithm is quite simple; output of NN is evaluated against desired output. If results are not satisfactory, bond between layers are modified and process is repeated again until fault is small enough. The objective of BP method is adapting synaptic credence in order to minimize a fault function. The approach most commonly used for the minimization of the fault function is based on the gradient method. The fault function is defined by Eq. (4).

IJASRM

International Journal of Advanced Scientific Research and Management, Volume 3 Issue 11, Nov 2018

www.ijasrm.com

ISSN 2455-6378

$$F(t)=\frac{1}{2}|e(t)|^2 \quad \dots (4)$$

where $e(t)$ is the fault value, i.e. the difference between the output and the estimated output. This difference is used to change the bond credence between neuron in the network according to Eq. (5).

$$\Delta w_j(t+1)=\eta\left(\frac{F(t)}{w_j}\right)+\alpha\,\Delta w_j(t) \dots (5)$$

where η is the learning rate stricture and the positive constant α is an impetus factor. Neural networks are sensitive to the choice of the learning rate and the impetus value. The learning rate can limit or expand the extent of Credence alteration in a learning cycle. A higher learning rate can make the network unstable and can cripple the network prediction ability. In the opposite if the learning rate is low, the training time of the network is increased. In the other hand, low impetus causes Credence wavering and inconsistency and thus neglecting the network from learning. High impetus can cripple the network adaptability. During the middle of training, when steep slope occurs, a small impetus value is recommended. However, during the end of training, large impetus value is desirable. Therefore, for the best results, it is necessary that these two strictures should be adjusted adaptively during the learning process, since no single value is optimal for all dimensions. In the next part, the implementation of fuzzy control system to adaptively determine the learning rate and impetus value is discussed.

3.3. Fuzzy logic based deep neural network

A fuzzy logic is a technique useful in representing human knowledge in a specific domain of application and in reasoning with that knowledge to make useful inferences or actions. A fuzzy logic control system consists of four components. A fuzzifier converts data into fuzzy data or Membership Functions (MFs). The fuzzy rule base contains the relations between the input and output. The fuzzy inference process combines MFs with the control rules to derive the fuzzy output, and the defuzzifier converts the fuzzy numbers back to a crisp value. There are two reasons that fuzzy logic systems are preferred: fuzzy systems are suitable for uncertain or approximate reasoning and they allow decision making with estimated values under in complete or uncertain information. By employing a fuzzy control system to adaptively adjust the learning strictures of the neural network according to the MSE fault, which can reduce the possibility of overshooting during the learning process and help the network get out of a local minimum. There are four strictures used to create the rules for the fuzzy logic control system; the supportive fault (SF), varying supportive fault (VSF), mark vary in fault (MVF) and embarrass summation of mark vary infault (ESMVF). The four strictures are described as follows:

$$\begin{cases} SF(t) = F(t)-F(t-1) \\ VSF = SF(t)-SF(t-1) \\ MVF(t)=1-\left\|\frac{1}{2}\left[mark(SF(t-1)+mark(SF(t))\right)\right\| \\ ESMVF = \sum_{m=t-4}^{t} MVF(m) \end{cases}$$

… (6)

For simplicity, we assume that the fuzzy logic system contains two inputs SF and VSF, and two output; the vary in the learning stricture Δ_η and vary in the impetus value Δ_α. The range of SF and VSF is 0–1.The linguistic values of SF, VSF and Δ_η are NL, NS,ZE, PS and PL. NL represents a ‘‘Negative Large” value, NS is ‘‘Negative Small”, ZE is ‘‘Zero”, PS is ‘‘Positive Small” and PL is ‘‘Positive Large”. The vary in the impetus value is small enough to avoid overcorrection. Tables 1 and 2 describe the milieu depictions of the rule bases for Δ_η and Δ_α, respectively.

Table 1. Decision table for Δ_η when $ESMVF \le 2$.

VSF	RE				
	NL	NS	ZE	PS	PL
NL	NS	NS	NS	NS	NS
NS	NS	ZE	PS	ZE	NS
ZE	ZE	PS	ZE	NS	ZE
PS	NS	ZE	PS	ZE	NS
PL	NS	NS	NS	NS	NS

IJASRM

International Journal of Advanced Scientific Research and Management, Volume 3 Issue 11, Nov 2018

www.ijasrm.com

ISSN 2455-6378

Table 2. Decision table for Δ_α when $ESMVF \le 2$.

VSF	RE				
	NL	NS	ZE	PS	PL
NL	-0.01	-0.01	0	0	0
NS	-0.01	0	0	0	0
ZE	0	0.01	0.01	0.01	0
PS	0	0	0	0	-0.01
PL	0	0	0	-0.01	-0.01

3.4. Improved Fuzzy Deep Learning Network training

Deep multi-layer neural networks have many levels of non-linearity allowing them to compactly represent highly non-linear and highly varying functions. The training phase of deep neural network contains two major steps of stricture initialization and improvement. The initialization step is critical in deep learning because the whole learning system is not convex. A better initialization strategy may help the neural network to converge to a good local minimum more efficiently. In this paper, a LIE model is introduced. The improvement step allows to precisely adjusting the strictures in the neural network in a supervised way to enhance the discriminate ability of the final feature presentation. To conduct this step, we propose to use the back-propagation method with gradient based optimization technique and an on-line fuzzy logic controller in the role of supervised to acclimatize the learning strictures based on the mean squared fault generated by the deep neural network. The training procedure can be summarized by Algorithm 1.

Algorithm 1.

Given the training sample and the structure of the deep neural network.

Step 1: *StricturesInitialization*

1. *Train the first layer as an instincitve codifier with the training sets as the input using the back – propagation algorithm.*
2. *Train the second layer as an instinctive codifier taking the first layer's outputas input.*
3. *Iterate as in* 2 *for the desired number of layers.*

Step 2: Im *provement*

4. *Use the output of the last layer as the input for theclassification layer, and initialize its strictures by supervised training.*
5. Im *prove the strictures of all layers with the Back – propagation method in a supervised way using fuzzy logic controller to adaptively adjust the learning strictures.*

4. Results and Discussions

DNS traffic generally uses User Datagram Protocol (UDP) or Transmission Control Protocol (TCP) on port 53 for doing the task of communication. Many of the DNS communication activities are conducted by making use of UDP since many of the protocol resolvers utilize it by defaulting.Earlier, TCP has been utilized zone transfers, other than that of RFC 1123 [9] expanded the use of TCP as a backup communication protocol when the answer needs to be larger than 512 octets. With this connection, the first UDP DNS response has only fractional answers. The truncation bit is set with the intention that the resolver possibly replicates the query over TCP. On the other hand, RFC 2671, "EDNS0" [10], defined a new opcode field / pseudo resource record that allows UDP DNS traffic to be bigger than 512 octets. This is due to roughly all of today's DNS traffic uses UDP as its transport protocol. Sorting the obtained records by different criterion is used to detect unusual records or activities. At the same time as searching for records with low TTL values can generally be useful in detection of fast flux domains. The top 10 wireless client IP addresses and the details are shown in Table 1. Table 2 depicts the details which consist of queries per day, domain name, number of anomaly received requests and number of failed requests about the top 10 destination domains.

IJASRM

International Journal of Advanced Scientific Research and Management, Volume 3 Issue 11, Nov 2018

www.ijasrm.com

ISSN 2455-6378

Table – 1: Top 10 wireless client IP addresses

Client S. No.	IP Address	Queries / Day	Number of Anomaly Requests Sent	Number of Failed Requests
1	139.59.17.152	2,41,056	40,338	10,650
2	59.99.150.200	2,28,159	19,436	8,080
3	223.179.220.177	1,14,311	12,719	4,412
4	118.151.209.5	95,981	9,736	3,417
5	203.90.4.145	84,452	7,642	3,062
6	103.234.190.183	74,191	5,618	3,391
7	122.176.20.6	73,445	3,749	3,234
8	150.107.25.248	65,049	5,085	2,545
9	112.133.201.135	62,237	4,896	2,947
10	103.1.115.219	52,617	3,827	1,865

Table – 2: Top 10 destination domain names

Domain S. No.	Domain Name	Queries / Day	Number of Anomaly Received Requests	Number of Failed Requests
1	baidu.com	37,89,755	1,02,749	2,23,080
2	qq.com	26,61,189	91,836	1,71,689
3	akadns.net	23,87,316	84,782	1,53,196
4	apple.com	17,90,093	54,736	1,02,531
5	taobao.com	11,86,935	61,038	67,259
6	in-addr.arpa	10,89,644	64,954	67,533
7	google.com	9,81,895	92,743	67,197
8	weibo.com	7,98,301	90,664	47,710
9	sina.com.cn	7,85,648	86,498	49,648
10	360.cn	7,55,655	85,053	43,928

Table – 3: Detection Accuracy of Anomaly Request at Client Side

	PCA - SVM Classifier					MGA-IFELM					FMLDNN				
Client No.	TP	TN	FP	FN	Accuracy	TP	TN	FP	FN	Accuracy	TP	TN	FP	FN	Accuracy
1	15301	14994	5045	4998	75.10	22008	14278	1021	3031	89.95	22358	15002	704	2274	92.62
2	8521	5783	2221	2911	73.60	11174	6129	1016	1117	89.03	11494	6468	733	741	92.42
3	5857	3809	1721	1332	76.00	7088	4366	614	651	90.05	7229	4618	518	354	93.14
4	4901	2592	1209	1034	76.96	5735	3001	477	523	89.73	5937	3203	297	299	93.88
5	3671	2201	994	776	76.84	4356	2506	481	299	89.79	4481	2618	377	166	92.89
6	2193	2114	712	599	76.66	2710	2392	255	261	90.82	2859	2488	127	144	95.18
7	1351	1389	418	591	73.09	1868	1483	181	217	89.38	1938	1511	121	179	92.00
8	2125	1759	516	679	76.47	2805	1781	199	300	90.19	3005	1803	111	166	94.55
9	2217	1537	493	649	76.67	2980	1438	191	287	90.24	3114	1533	118	131	94.91
10	1548	1381	502	396	76.54	2549	908	153	216	90.36	2640	1009	81	96	95.37

IJASRM

International Journal of Advanced Scientific Research and Management, Volume 3 Issue 11, Nov 2018

www.ijasrm.com

ISSN 2455-6378

Table – 4: Detection Accuracy of Anomaly Request at Server Side

	PCA - SVM Classifier					MGA-IFELM					FMLDNN				
Client No.	TP	TN	FP	FN	Accuracy	TP	TN	FP	FN	Accuracy	TP	TN	FP	FN	Accuracy
1	52894	26052	12054	11749	76.83	61902	28993	6537	5317	88.46	64987	30874	4048	2840	93.30
2	34794	34191	11892	10959	75.12	46883	33826	5701	5426	87.88	49996	35819	3011	3010	93.44
3	41046	21449	10945	11342	73.71	52470	22646	5527	4139	88.60	54027	23768	3396	3591	91.76
4	21562	19093	7934	6147	74.27	30444	18463	3418	2411	89.35	32854	19509	1217	1156	95.66
5	22703	23503	7945	6887	75.70	31622	22228	4483	2705	88.22	33538	23244	2008	2248	93.03
6	26566	23543	7933	6912	77.15	36522	21846	5201	1384	89.86	38957	23113	1989	894	95.56
7	35748	35428	8844	12723	76.75	47132	35702	2491	7418	89.32	49856	37291	1479	4117	93.97
8	31095	37846	12034	9689	76.04	41078	39184	4454	5948	88.53	43639	41094	3286	2645	93.46
9	37125	29642	9948	9783	77.19	46417	32164	2401	5516	90.85	48771	34166	1938	1623	95.88
10	35023	30543	10339	9148	77.09	46161	30008	5403	3481	89.55	48567	31997	2311	2178	94.72

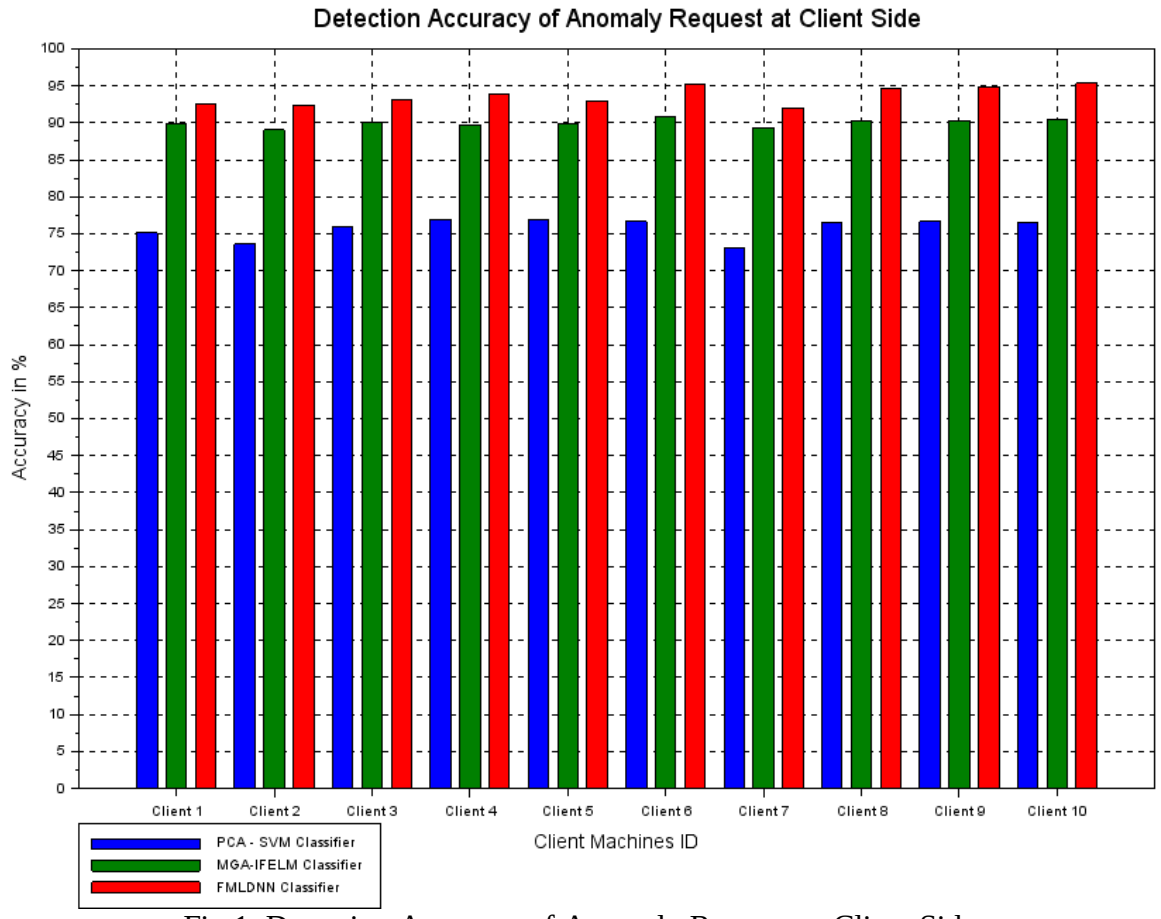

Fig.1. Detection Accuracy of Anomaly Request at Client Side

IJASRM

International Journal of Advanced Scientific Research and Management, Volume 3 Issue 11, Nov 2018

www.ijasrm.com

ISSN 2455-6378

Table – 5: Time Taken for Anomaly Detection – Client side

S.No.	IP Address	Queries / Day	Classification		
			PCA - SVM Classifier	MGA-IFELM	FMLDNN
1	139.59.17.152	2,41,056	21.70	9.57	6.05
2	59.99.150.200	2,28,159	20.53	8.93	4.11
3	223.179.220.177	1,14,311	10.29	3.85	1.89
4	118.151.209.5	95,981	8.64	3.10	1.42
5	203.90.4.145	84,452	7.60	3.66	1.18
6	103.234.190.183	74,191	6.68	2.62	0.89
7	122.176.20.6	73,445	6.61	2.82	0.61
8	150.107.25.248	65,049	5.85	2.44	0.43
9	112.133.201.135	62,237	5.60	2.48	1.16
10	103.1.115.219	52,617	4.74	2.01	0.99

Table – 6: Time Taken for Anomaly Detection – Server side

S.No.	IP Address	Queries / Day	Classification		
			PCA - SVM Classifier	MGA-IFELM	FMLDNN
1	139.59.17.152	2,41,056	341.08	83.16	44.37
2	59.99.150.200	2,28,159	239.51	64.45	31.11
3	223.179.220.177	1,14,311	214.86	59.75	31.84
4	118.151.209.5	95,981	161.11	51.29	29.16
5	203.90.4.145	84,452	106.82	29.11	18.69
6	103.234.190.183	74,191	98.07	27.39	11.28
7	122.176.20.6	73,445	88.37	25.99	12.01
8	150.107.25.248	65,049	71.85	16.36	10.12
9	112.133.201.135	62,237	70.71	12.96	6.01
10	103.1.115.219	52,617	68.01	11.95	5.98

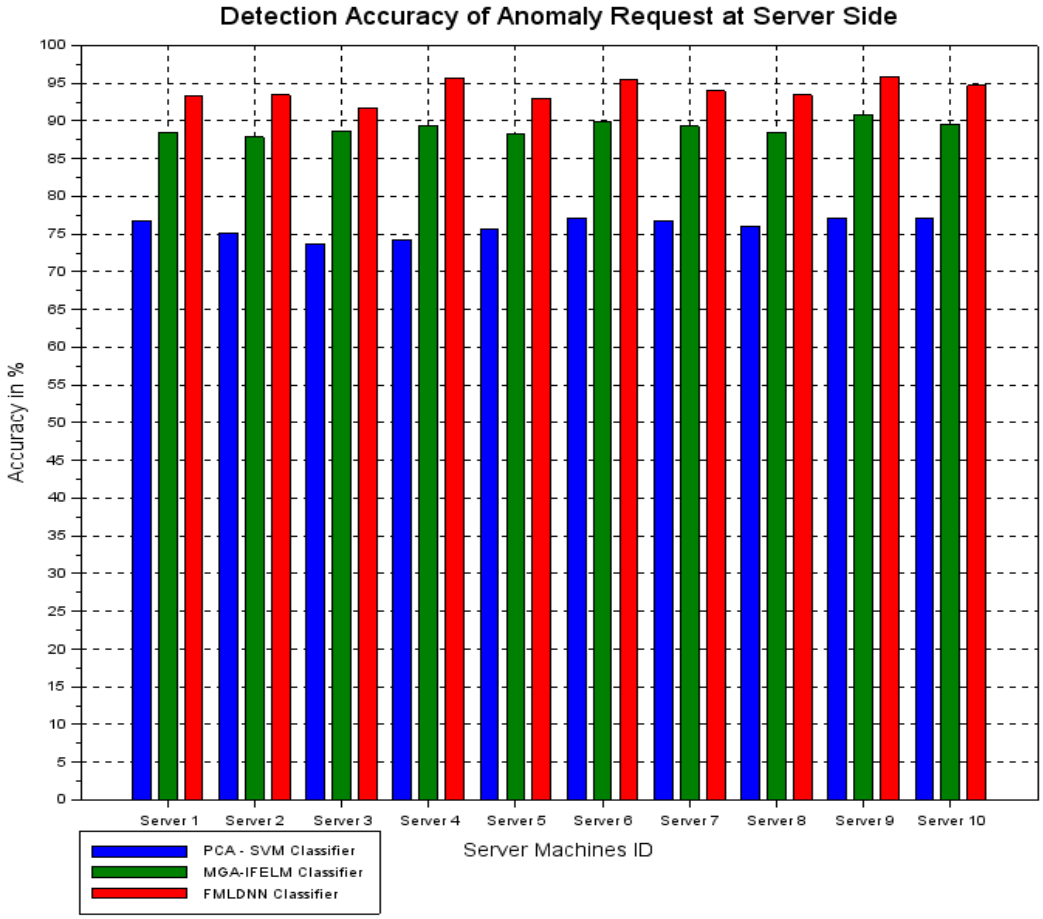

Fig.2. Detection Accuracy of Anomaly Request at Server Side

IJASRM

International Journal of Advanced Scientific Research and Management, Volume 3 Issue 11, Nov 2018

www.ijasrm.com

ISSN 2455-6378

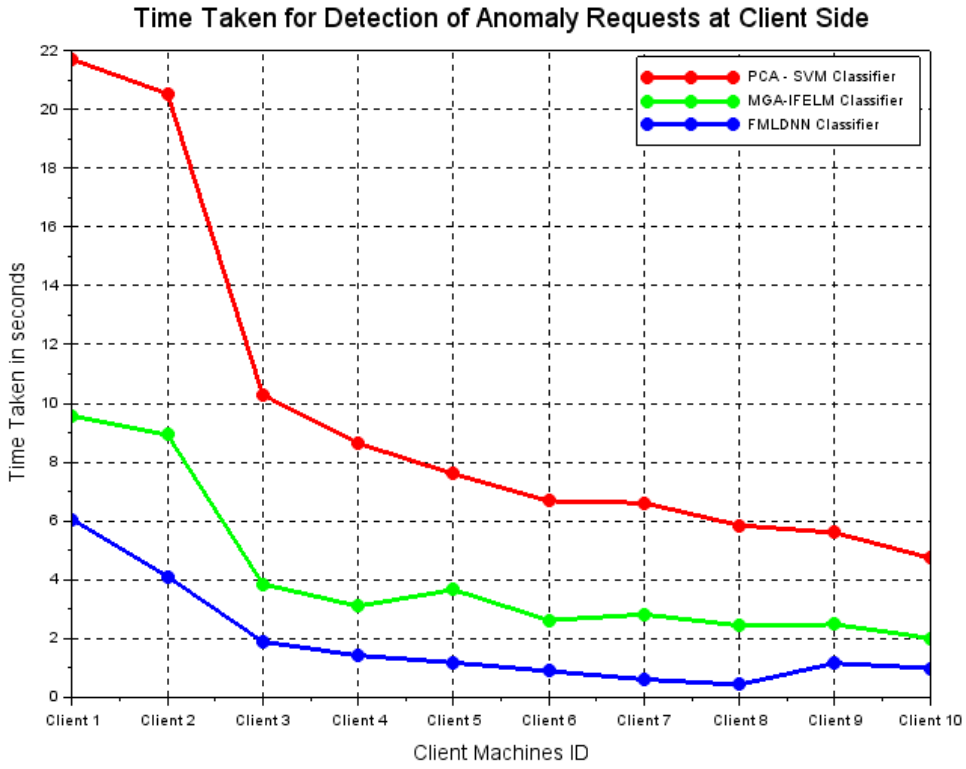

Fig.3. Time Taken for Anomaly Detection – Client side

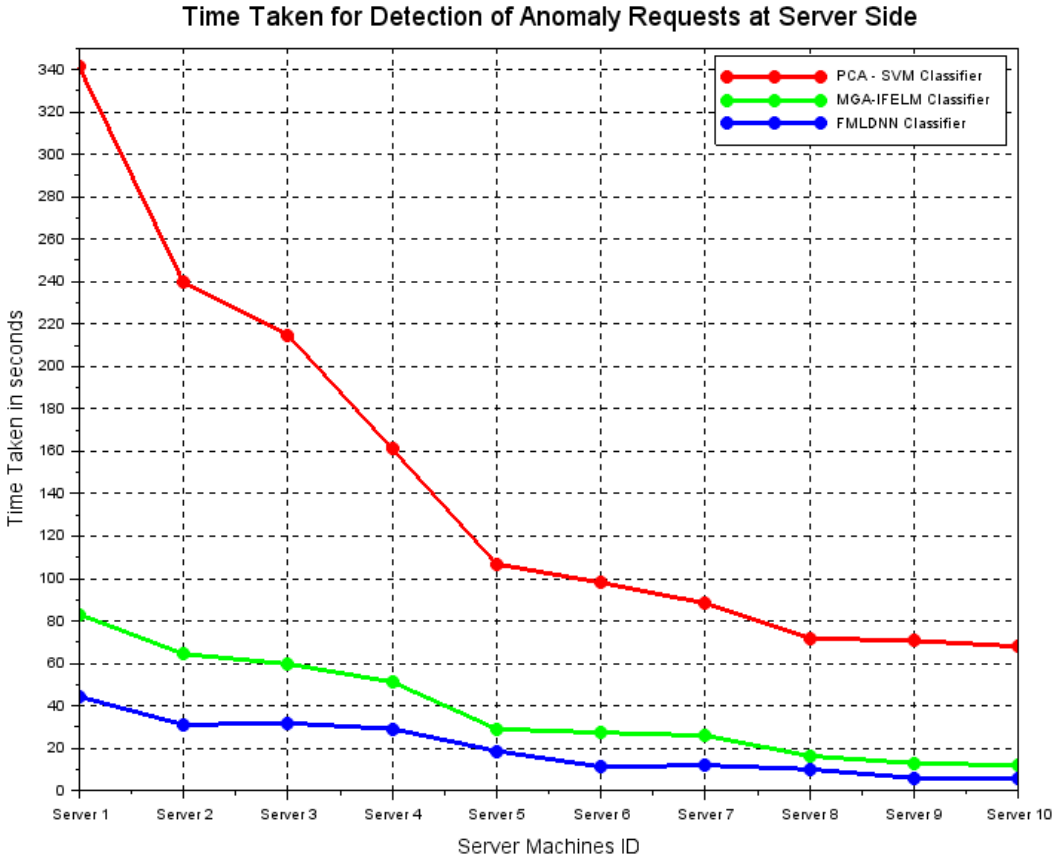

Fig.4. Time Taken for Anomaly Detection – Server side

Table – 3 presents the performance in terms of detection accuracy of anomaly requests at client side. Ten client machines are taken for consideration. PCA - SVM, MGA-IFELM [11] and FMLDNN classifiers are compared. It is clearly observed that FMLDNN classifier obtained better accuracy than that of PCA – SVM and MGA – IFELM. Also, it is to be noted that the classification of detection accuracy attained more than 90% at all the client machines with the maximum of 95.18%. This shows the robustness of FMLDNN in terms of anomaly detection. The bar chart results are presented in Fig.1. Table – 4 presents the performance in terms of detection accuracy of anomaly requests at server side. Like above, the proposed FMLDNN classifier is compared with PCA – SVM and MGA – IFELM. From the results it is clear that FMLDNN obtains better classification accuracy in detection of anomaly requests above 90% at all server machines with the maximum of 95.88%. The bar chart results are presented in Fig.2. Table – 5 portrays the performance in terms of time taken for the classification of anomaly detection at client side and Table – 6 magnifies the performance in terms of time taken for classification of anomaly detection at server side. From both the obtained results it is evident that FMLDNN consumes lesser time when compared with both PCA – SVM and MGA – IFELM. Fig. 3 and Fig. 4 shows the comparative analysis using line chart for the performance in terms of time taken for anomaly detection classification.

5. Conclusions

For the past several research works, classification of DNS anomaly requests in both server side and client side stands a big research problem in web mining. This research work aimed to proposed artificial neural network with fuzzy logic in order to perform the classification task of anomaly detection requests. The results are promising. The accuracy and time taken are the major performance metrics used in this research work to evaluate the effectiveness of the classifier. Results envisioned that FMLDNN performed better than that of previous conducted research works.

References

[1] A. Berger, W.N. Gansterer, Modeling DNS agility with DNSMap, in: Proceedings of IEEE INFOCOM Workshop on Traffic Monitoring and Analysis, Turin, Italy, 2013.

[2] X. Ma, J. Zhang, Z. Li, J. Li, J. Tao, X. Guan, J.C.S. Lui, D. Towsley, "Accurate DNS query characteristics estimation via active probing," Journal of Network and Computer Applications, vol. 47, pp. 72 - 84, 2015.

[3] Q. Lai, C. Zhou, H. Ma, Z. Wu, S. Chen, "Visualizing and characterizing DNS lookup behaviors via log-mining," Neurocomputing, vol. 169, pp. 100 - 109, 2015.

[4] Y. Jin, M. Tomoishi, N. Yamai, "A Secure and Lightweight IoT Device Remote Monitoring and Control Mechanism Using DNS," IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), pp. 282-283, 2017.

[5] B. S. Fagin, B. Klanderman, M. C. Carlisle, "Making DNS Servers Resistant to Cyber Attacks: An Empirical Study on Formal Methods and Performance," IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), pp. 566-571, 2017.

[6] N. F. Huang, C. C. Li, C. H. Li, C. C. Chen, C. H. Chen and I. H. Hsu, "Application identification system for SDN QoS based on machine learning and DNS responses," 19th Asia-Pacific Network Operations and Management Symposium (APNOMS), pp. 407-410, 2017.

[7] R. Romero-Gomez, Y. Nadji, M. Antonakakis, "Towards designing effective visualizations for DNS-based network threat analysis," IEEE Symposium on Visualization for Cyber Security (VizSec), pp. 1-8, 2017.

[8] M. S. Manggalanny, K. Ramli, "Combination of DNS traffic analysis: A design to enhance APT detection," 2017 3rd International Conference on Science and Technology - Computer (ICST), pp. 171-175, 2017.

[9] Conrad Albrecht-buehler, Benjamin Watson, David A Shamma, Visualizing live text streams using motion and temporal pooling. IEEE Comput. Graph. Appl. 25 (3) (2005) 52–59.

[10] Ulrik Brandes, Daniel Fleischer, Jrgen Lerner, Highlighting conflict dynamics in event data, in: IEEE Symposium on Information Visualization, 2005. INFOVIS 2005. IEEE, 2005. New York, NY, USA, pp. 103–110.

[11] Pieter Lexis, Matthijs Mekking, Identifying Patterns in DNS Traffic, 2013.

[12] H. Yu, et al., A visualization analysis tool for DNS amplification attack, in: 2010 3rd International Conference on Biomedical Engineering and Informatics (BMEI), IEEE, New York, NY, USA, 2010, pp. 2834–2838.

[13] Kenton Born, David Gustafson, Ngviz: detecting dns tunnels through n-gram visualization and quantitative analysis, in: Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research, ACM, New York, NY, USA, 2010, p. 47.

[14] Euihyun Jung, Joonhyung Lim, Juyoung Kim, An Analysis of the Korea National DNS Using Big Data Technology. Frontier and Innovation in Future Computing and Communications, Springer, Netherlands (2014) 605–613.

[15] W.Z. Ruan, Y. Liu, R.L. Zhao, Pattern discovery in DNS query traffic, Proc. Comput. Sci. 17 (2013) 80–87.

[16] David Dagon, Large-scale DNS data analysis, in: Proceedings of the 2012 ACM Conference on Computer and communications Security (CCS '12), ACM, New York, NY, USA, pp. 1054–1055.

[17] Mark Allman, Thomas Callahan, Michael Rabinovich, On modern DNS behavior and properties, SIGCOMM Comput. Commun. Rev. 43 (3) (July 2013) 7–15.

[18] Internet Engineering Task Force: Requirements for Internet Hosts Application and Support. RFC 1123 (October 1989).

[19] Vixie, P.: Extension Mechanisms for DNS (EDNS0). RFC 2671 (August 1999).

[20] Dr.V.Kathiresan S.S.Suganya, Modified Genetic Algorithm based Improved Fuzzy Extreme Learning Machine (MGA-IFELM) for Anomaly Detection in DNS Query Logs, International Journal of Pure and Applied Mathematics, 2018/5