I.J. Wireless and Microwave Technologies, **2022, 6, 25-32**
Published Online on December 8, 2022 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2022.06.03

Pure-Octet Extraction based Technique for Identifying Malicious URLs based on IP Address Attributes

Aasha Singh*
KNIT, Sultanpur, India
E-mail: researchcse19@gmail.com
ORCID iD: https://orcid.org/0000-0001-9002-1494
*Corresponding Author

Awadhesh Kumar
KNIT, Sultanpur, India
E-mail: awadhesh@knit.ac.in
ORCID iD: https://orcid.org/0000-0001-9055-2500

Ajay Kumar Bharti
BBDU, Lucknow, India
E-mail: ajay_bharti@hotmail.com
ORCID iD: https://orcid.org/0000-0001-6879-5151

Vaishali Singh
MUIT, Lucknow, India
E-mail: singh.vaishali05@gmail.com
ORCID iD: https://orcid.org/0000-0001-8304-8947

Received: 19 May, 2022; Revised: 05 June, 2022; Accepted: 27 July, 2022; Published: 08 December, 2022

Abstract: On the basis of characteristics derived from IPv4 addresses, this paper offers a method for identifying interaction linked with website-based malware and then modelling a machine-learning-based classifier. In this research work, a modified approach is proposed for detecting fraudulent websites and compared with other methods like SVM assessment of IP addresses, octet-based technique, modified extended version of octet-based technique, and bit string-based characteristics. This modified approach is based on the fact that logical addressing is more reliable and consistent than other measures like URLs and DNS. The characteristic sequence which makes up URLs and domain names are more changeable with respect to IP addresses which are less changeable in comparison to URLs or domain names. The IPv4 address length is encoded into 4-byte space. Here, we have evaluated our modified approach with valid IP addresses from Kaggle [11], published on January 16, 2018, have been used to validate the efficacy of their method.

Index Terms: URLs, SVM, IP address, Malware, Pure-Octet, Decision Tree, Accuracy.

1. Introduction

Malware attacks, specifically web-based, have evolved into one of the most significant dangers that requires immediate attention where a user's confidential and personal information is compromised, or some malicious software is forcefully downloaded at the client side. Such attacks exploit flaws in web browsers and plugins like Java virtual machines, PDF plugins etc.

Malware authors disguise malware propagation sites as interesting and enjoyable themes on social media websites or in e-mails to trick people into visiting them. To avoid discovery, they use obfuscation techniques like cryptography and tunnelling. For one-time access, several intermediary redirecting URLs are effective. Using different blacklists is one of the techniques that have gained attention as possible ways of identifying such viruses. Due to the flexibility of

malicious websites, such techniques frequently fail to identify new assaults. As a result, keeping blacklists containing information about new dangerous websites is challenging.

While modifying the algorithm and results shown in the paper [13], we conducted several experiments that suggest that the overall process can be simplified if the feature selection is restricted to only the first two octets (first 16 bits) of the IP addresses and their combination in the Octet-based feature extraction method, before training the machine learning based model.

We have implemented "Decision Tree Classifier" machine learning algorithm in place of "Support Vector Machines", which is utilized in the paper [13], for an improved result in our proposed work. However, it is to be noted that if our suggested model is trained on a sufficiently larger dataset, such as an instantaneous data on an institute intranet used in the paper [13], the results may vary consequently.

2. Literature Survey

Blacklists malicious identification and Reputation systems, client honeypots, and Intrusion Detection Systems (IDS), are some types of systems suggested in this domain of detecting malicious websites. The most common techniques for preventing users from visiting harmful websites are blacklists or systems based on reputations of a website. These techniques can be used for both host and consumer-side filtering. Reflection is built on several sorts of data found in IP addresses, and it is used to keep people from visiting dangerous websites. Features obtained from domains and link topologies are among the reputation criteria which evaluated the performance of websites security system.

Antonakakis et al. [1] created the "Notos" adaptive reputation system. It gathers data from a variety of sources in order to mimic the network behaviours of both benign and malicious websites. Then, for each URL, it uses these methods to compute a score. Felegyhazi et al. [3] advocated dynamic blacklist based on hostname. On the basis of registry and server statistics, it uses a few collection of well-known harmful area to anticipate dangerous domains. Based on both semantic architecture and host-based characteristics, Ma et al. [4] developed a supervised learning technique for categorising webpages as benign or harmful. Renjan et al. [5] proposed Dynamic Attribute-based Reputation (DAbR), an approach for generating reputation ratings for IP addresses using existing data known to be problematic.

Users can be blocked from visiting dangerous websites using Intrusion Detection Systems (IDS). IDS like Bro [6] and Snort [7], observe the system for signals that match predetermined attack patterns. Fingerprint-based IDS cannot identify new threats since threat fingerprints are produced by known threats. Irregularity-based IDS understands conventional pattern and utilises it to identify intrusions. As a result, it has the ability to identify unexpected assaults [8]. However, there's a good chance that irregularity-based IDS would mistakenly identify typical data as an attack, resulting in wrongful convictions.

Drive-by-download assaults are detected and evaluated using client honeypots [2], [9], [10]. A consumer-side honeypot is a software that scans the internet for dangerous webpages and discovers them. Limited honeypots and large honeypots are the two varieties of customer-side honeypots. Honeypots with low involvement, such as HoneyC [9], contain a desktop emulation for crawling webpages. As a result, there is no chance of them becoming malware infected. Larger honeypots, such as Capture-HPC [10], contain a genuine internet browser and operating system, allowing them to collect additional details after infection, such as virus activity. Marionette, a high interaction honey trap suggested by

Akiyama et al. [2], eliminates the danger of malicious attacks. Marionette has a true sensitive internet browser, as well as extensions that may identify assaults without infecting it with viruses. The conventional approaches to detect malicious websites frequently break down to identify new attacks as a consequence of their novelty and flexibility. To handle this problem, our research work offers a technique for identifying malicious websites/ URLs based on IP address attributes. Here we are proposing a modified approach, in contrast to the one mentioned in the paper [13], to obtain an improved result.

Authors in [14], analyzed the various machine learning algorithms for preventing e-mails from the spams using spam classifications and filtering techniques. In [15], authors proposed a model to prevent e-mails from spams using artificial neural network. They have achieved 7.5% nearer to XEAMS system for anti-spam. Author in [16], proposed a combined model for detecting the e-mail spam using particle swarm optimization and artificial neural network algorithms for feature extraction and to detect spam they applied SVM classifier. In [17], authors have proposed a technique to disclose the intrusions using support vector regression classifier on a standard dataset. They have applied feature extraction methods for filtering the dataset before training and testing procedure. They have achieved satisfactory results as their experimental performance for accuracy.

3. Octet-based Extraction Method

According to Octet-based Extraction method described in the paper [13], a feature vector is mathematically formulated by the following equation (1):

$$\begin{cases} b_k = 1 \;\; (k \;\; in \;\; \cup_{n=1}^{N}(2^8 \cdot (n-1) + X_n)) \\ b_k = 0 \;\; (otherwise), \end{cases} \quad (1)$$

3.1 Pure-Octet based Extraction

While modifying the approach and improving the results claimed in the paper [13], we experimented with various combinations of the existing and extracted features. The results suggest that the overall process can be simplified if the feature selection is restricted to only the first two octets of the IP addresses and their combination in the Octet-based feature extraction method, before training the machine learning based model. The first two octets majorly help to predict whether the website would be malicious or benign. This is primarily because they represent the network to which they belong. The last octet represents the host that no considerable effect on the final output. We call it ***"Pure Octet based Extraction."***

3.2 Ex-Octet based method

The ***Ex-Octet method*** further extrapolates Octet's feature vector to formulate an $M = 2^8 \times (N + 2)$ – dimensional feature vector denoted as a scarce bit sequence $\{b_0, \ldots, b_{m-1}\}$ from the utmost major N octets of an IPv4 address, where N is a natural number larger than or equivalent to three. The preliminary value for each bit $\{b_0, \ldots, b_{m-1}\}$ is zero [13]:

$$\begin{cases} \left(k \;\; in \;\; \cup^{N} \;\; \{2^8.m + (\sum_{i=1}^{m-1} Xi \,) mod.\, 2^8\}\right) \\ b_k = 0 \;\; (otherwise), \;\; M = N \geq 3 \end{cases} \quad (2)$$

From our observations to include only the first two octets X_1 and X_2, we further included the extended dimension of (N+2) - dimensional feature using only the same octets which is implemented as a $[k = 2^8 \text{ x } 4 + (X_1 + X_2 + X_3) \bmod 2^8]$ vector which gave us consistent results as the overall accuracy improved marginally. We observed that Octet based feature extractions have the same impact on the overall accuracy as purely using octets without complex mathematical manipulations. However, as a mathematical modification to the approach, we also tried a new (N+3) dimensional feature implemented as $[k = 2^8 \text{ x } 5 + (X_1 + X_2 + X_3 + X_4) \bmod 2^8]$ but it did not show any improvement in results.

3.3 Machine Learning Classifiers

We have implemented "Decision Tree Classifier" or "Random Forest Classifier" as machine learning algorithm in place of "Support Vector Machines", for better improvement of results. In its most basic form, decision trees are a resolution tool that employs a root and branch-like architecture to discover potential outcomes via predicates. The most used supervised learning approaches use a tree-based algorithm in today's era. As such, a basic model may be understood as models with high relative precision, durability, and usability.

3.3.1 Improved Accuracy

SVM requires several models for classification of more than one classes, but Decision Tree does not. It has the capacity to provide a probability over the forecast, while SVM cannot perform this task. It also does a decent job handling classification data than SVM. Decision Tree Classifier performed with an overall improved accuracy over Support Vector Machines algorithm, as implemented in the paper.

3.3.2 Lower Computational Complexity

As we can see the Table1, different machine learning algorithms are given with their Time complexities for the classification/regression, Training and Prediction.

Table 1. Time Complexity of SVM, Decision Tree, and RFC

Algorithm	Classifcation/Regression	Training	Prediction
Decision Tree	C+R	$O(n^2p)$	$O(p)$
Random Forest	C+R	$O(n^2pn_{tress})$	$O(pn_{tress})$
Random Forest	R Breiman implementation	$O(n^2pn_{tress})$	$O(pn_{tress})$
Random Forest	C Breiman implementation	$O(n^2\sqrt{p}n_{tress})$	$O(pn_{tress})$
SVM (Kernel)	C+R	$O(n^2p + n^3)$	$O(n_{sv}p)$

Support Vector Machines (SVM) has a significantly higher time complexity than Random Forest Classifiers and Decision Trees, as seen in *Table 1*. This indicates that whenever the amount of the training data is larger, training a SVM model takes more time than training a Decision Tree, which is in our situation. This must be taken into account while selecting an algorithm. When amount of data given to a SVM classifier surpasses 20,000, it becomes less useful. As a result, as soon as the test sample gets larger, decision trees can be favoured.

However, it is to be noted that if our suggested model is trained on a sufficiently larger dataset, such as an instantaneous data on an institute intranet used in the paper [13], the results may vary consequently. Due to lack of a sufficiently large dataset, our method and changes will only reflect any actual improvement in results if the training data is scaled up.

4. Proposed Methodology

The following Fig.1 captures the overview of our method in a block diagram. Our proposed methodology takes final dataset as malicious IP addresses as well as benign URLs. After applying feature extraction methods, we found reduced malicious feature vectors as well as benign feature vectors as final dataset. After that decision tree is applied for the result.

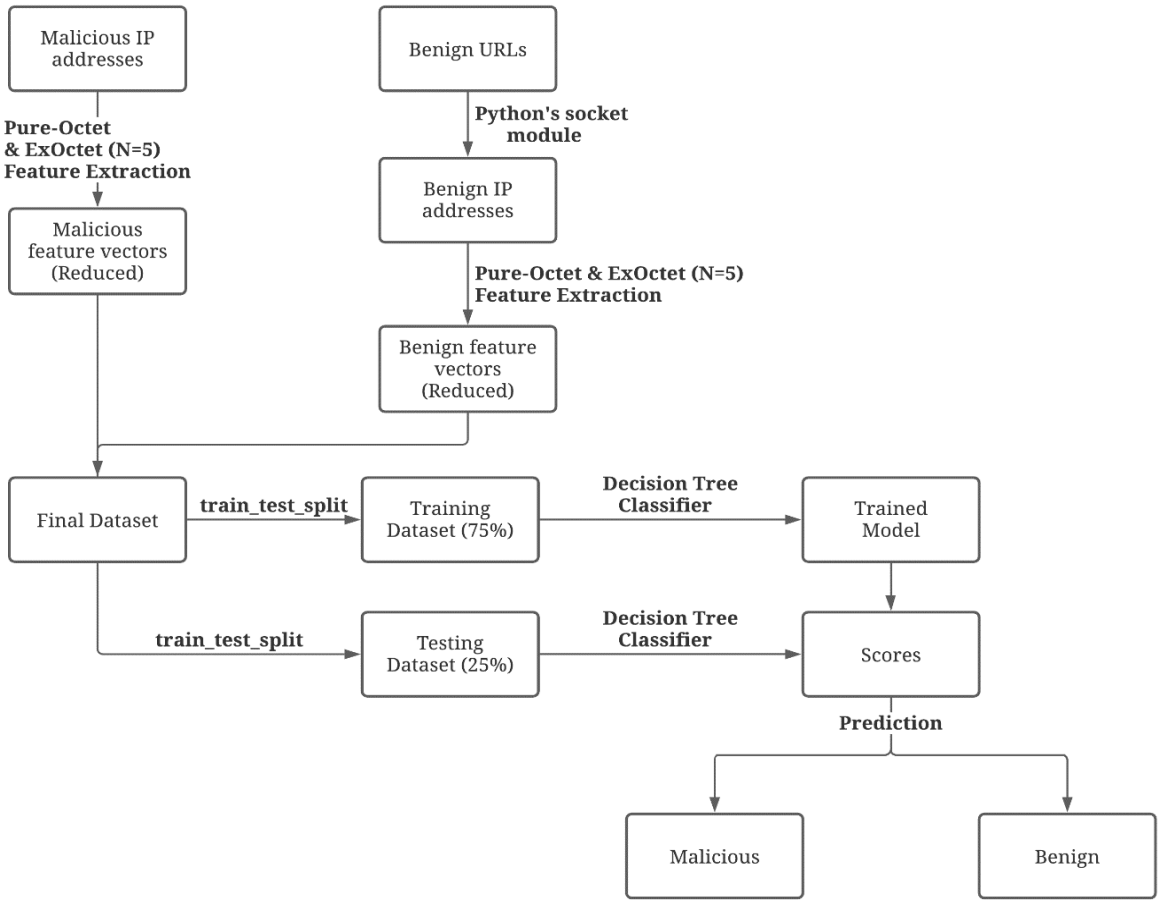

Fig. 1. Proposed methodology for improving accuracy

5. Experimental Setup

We have implemented our proposed methodology using Python Language (v3.7.4) and using the libraries: NumPy and pandas for data pre-processing, and data visualisation purpose we have also used matplotlib and seaborn, as well as machine-learning algorithms and metrics the sklearn is used. The project can be executed on both PyCharm (using the python executable or PY file) as well as Jupyter Notebook (using the IPYNB file).

5.1 Dataset applied for Methodology

We acquired IP addresses of benign and malicious websites to test our technique. The data we gathered for creating the trained model and evaluating our technique is shown in Table 2. The benign data is composed of URL addresses of more 70,000 websites sorted by their popularity on the Internet and is obtained from Kaggle [11], published on January 16, 2018. From these URLs, we resolved their corresponding IP addresses using Python's socket module. The malicious IP addresses are obtained from a malware IP blocklist [12], which contains 24,439 malicious websites. The dataset was further divided into train and test data using Python's pandas library. The training data comprises of 75% approx. of the overall data while the remaining 25% approx. of the rest part for the testing data.

Table 2. Division of Training and Testing data

	Malicious IPs	**Benign IPs**	**Total**	**Total %**
Train dataset	18 389	52 751	70 414	71.9
Test dataset	6 050	17 663	24 439	28.1
			94853	

5.2 Evaluation Measures

This step is the most important element of the study since it addresses the query of whether or not the experiment was successful. The following should be completed in order to get the solution to this question:

(1) To select acceptable measures for assessing the accuracy of the model.
(2) To compare the specified measures to values from a previous study that was already a success.

The following two situations are anticipated as a result of the findings:

(1) The assumption fails – the chosen measures would deteriorate equivalent values in some different studies significantly.
(2) The assumption is valid – the value of the measures is comparable to values of other studies or even indicates an increase in performance. The essential findings can be stated regardless of the ultimate result.

True-Positive (TP), False-Positive (FP), True-Negative (TN), and False-Negative (FN) are the four fundamental metrics used to assess the effectiveness of the models [13].

- Explanation of these terms are as follows:

1) TP – classifier properly detects a malicious URL
2) FP – classifier inaccurately detects a malicious URL
3) TN – classifier properly detects a benign URL
4) FN – classifier inaccurately detects a benign URL

We can see a graphical depiction of these classifier metrics, and this is known as Confusion Matrix in Fig. 2.

		Predicate	
		Present	Not Present
Class	Class 2	True-Positive	False-Positive
	Class 1	False-Negative	True-Negative

Fig. 2. Confusion matrix representation for a classifier

Other measures provide a more comprehensive picture of a model's correctness. The Precision, Recall, and F1 Score computed from the fundamental metrics described above, are among these measures. Table 3 summarises these measures and their formulae.

Table 3. Evaluation metrics

	Measures	Formula
1	Accuracy	(TP+TN) / (TP+FP+FN+TN)
2	Precision (P)	TP / (TP+FP)
3	Recall (R)	TP / (TP+FN)
4	F1 Score	2 x (P x R) / (P+R)

6. Results and Discussions

As our methodology is implemented with given dataset, we found two classes like (i) benign and (ii) malicious of the applied dataset. The following section provides an insight to the performance of the model. We have calculated the overall accuracy, precision and recall values to measure the performance of our proposed model. These values for the best result obtained during our experiments are shown below in Table 4, which is obtained using classification report in Python's sklearn library.

Table 4. Values of metrics

Class	Precision	Recall	F1-score	Accuracy
0: benign	0.93	0.96	0.94	
1: malicious	0.86	0.80	0.83	
average/total	**0.79**	**0.86**	**0.82**	**0.915**

Table 4 indicates that our model's precision is 0.79, the value of recall is equal to 0.86, and the overall accuracy for the model is 0.915. The above data also can be mathematically acquired from the confusion matrix, shown in Fig. 3.

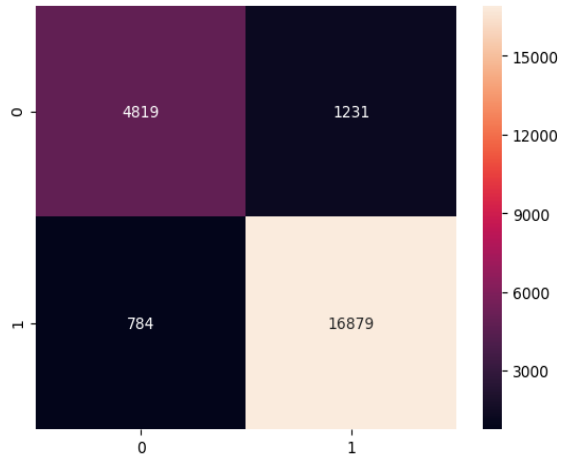

Fig. 3. Confusion Matrix for given dataset

For the result found during the experiment the ROC curve is plotted and it is shown in Fig. 4. As we know that the ROC curve is used to show the exchange in the midst of (sensitivity) and (1- specificity). Machine learning classifiers that offer curves nearer to the top-left corner point to a superior performance. As a reference point, the random classifier is supposed to provide points lie down next to the diagonal (FPR = TPR).

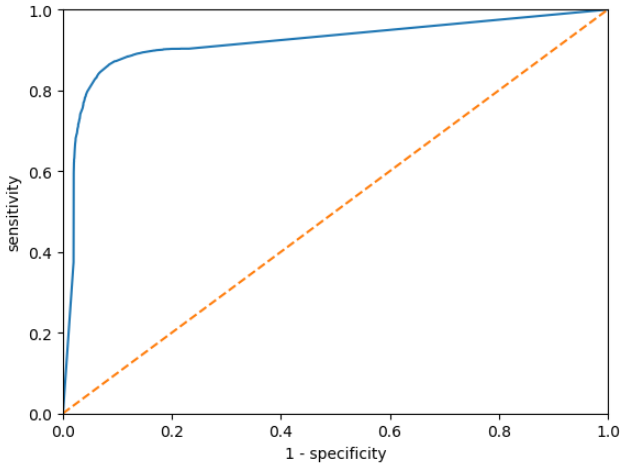

Fig. 4. ROC Curve for the best result

6.1 Performance and Result Comparison

Here, we have shown the results obtained from all our experiments and all these results are compared with each other. The comparison is put together and is presented in the *Table 5*.

Table 5. Result comparison

Approach	Algorithm	Category of the applied features	Accuracy (%)	Precision (%)	Recall (%)
Pure Octet Based	Random Forest	First two Octets (X1, X2)	91.37	79.45	85.67
Pure Octet Based	Decision Tree	First two Octets (X1, X2)	91.50	79.65	86.01
Pure Octet Based including (N+2) dimensional vector	Random Forest	First two octets (X1, X2) and an extended octet (N = 5)	91.18	79.07	85.29
Pure Octet Based including (N+2) dimensional vector	Decision Tree	First two octets (X1, X2) and an extended octet (N = 5)	91.38	79.45	85.7
Bit String Based	Random Forest	First 16 bits of IP	90.78	85.03	78.12
Bit String Based	Decision Tree	First 16 bits of IP	91.1	85.76	78.67
Bit String Based	Random Forest	First 24 bits of IP	88.42	78.01	76.98
Bit String Based	Decision Tree	First 24 bits of IP	89.91	90.12	68.55

7. Conclusions

In this paper, we have developed an understanding about different ways to classify websites as malicious or benign by learning their IPv4 address features and training a model depending on the different machine learning techniques. Our experiments indicate that features acquired directly from the IPv4 addresses, when limited to only the first two octets (16 bits) and their combination in the Octet-based feature extraction method can help improve the overall accuracy. This indicates that the size of malicious websites network is larger than that used in the previous works, as it gave best result on considering first 3-Octets (24 bits). We have used changed algorithm "SVM" to "Random Forest Classifier" or "Decision Tree" as they clearly give better results and have lower computational complexity. Our method could achieve a marginal improvement of 1% with an accuracy of 91.5% by reducing the feature set and a change in algorithm. However, it is to be noted that if our suggested method is modelled and trained on a sufficiently larger dataset, such as an instantaneous data on an institute intranet, the results may vary consequently. Due to difference in the dataset of our experiment, the Random Forest or Decision Tree algorithms may not provide further better results on the dataset used in the research paper. Also, the use of Decision Tree Classifier may result in an overfitted model that may not perform well when it is required to predict malicious IPs it has not encountered before.

References

[1] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, "Building a dynamic reputation system for dns," in Proceedings of the 19th USENIX conference on Security, ser. USENIX Security'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 18–18.

[2] M. Akiyama, M. Iwamura, Y. Kawakoya, K. Aoki, and M. Itoh, "Design and implementation of high interaction client honeypot for drive-by-download attacks," IEICE Transactions on Communications, vol. E93.B, no. 5, pp. 1131– 1139, 2010.

[3] M. Felegyhazi, C. Kreibich, and V. Paxson, "On the potential of proactive domain blacklisting," in Proceedings of the 3rd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more, ser. LEET'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 6–6.

[4] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious urls," in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ser. KDD '09. New York, NY, USA: ACM, 2009, pp. 1245–1254.

[5] A. Renjan, K. P. Joshi, S. N. Narayanan and A. Joshi, "DAbR: Dynamic attribute-based reputation scoring for malicious IP address detection", *IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 64-69, Nov 2018.

[6] V. Paxson, "Bro: a system for detecting network intruders in real-time," in Proceedings of the 7th conference on USENIX Security Symposium - Volume 7. Berkeley, CA, USA: USENIX Association, 1998, pp. 3–3.

[7] M. Roesch, "Snort - lightweight intrusion detection for networks," in Proceedings of the 13th USENIX conference on System administration, ser. LISA '99. Berkeley, CA, USA: USENIX Association, 1999, pp. 229–238.

[8] M. Ishida, H. Takakura, and Y. Okabe, "High-performance intrusion detection using optigrid clustering and grid-based labelling," in Applications and the Internet (SAINT), 2011 IEEE/IPSJ 11th International Symposium on, Jul. 2011, pp. 11 –19.

[9] C. Seifert, I. Welch, P. Komisarczuk et al., "Honeyc-the lowinteraction client honeypot," Proceedings of the 2007 NZCSRCS, Waikato University, Hamilton, New Zealand, 2007.

[10] Capture-hpc. [Online]. Available: https://projects.honeynet. org/capture-hpc/

[11] https://www.kaggle.com/cheedcheed/top1m/metadata

[12] http://lists.blocklist.de/lists/all.txt

[13] D. Chiba, K. Tobe, et al., "Detecting Malicious Websites by Learning IP Address Features", 2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet, 2012.
[14] Mahmoud Jazzar, Rasheed F. Yousef, Derar Eleyan, "Evaluation of Machine Learning Techniques for Email Spam Classification", International Journal of Education and Management Engineering (IJEME), Vol.11, No.4, pp. 35-42, 2021. DOI: 10.5815/ijeme.2021.04.04.
[15] Yahya Alamlahi, Abdulrahman Muthana,"An Email Modelling Approach for Neural Network Spam Filtering to Improve Score-based Anti-spam Systems", International Journal of Computer Network and Information Security(IJCNIS), Vol.10, No.12, pp.1-10, 2018.DOI: 10.5815/ijcnis.2018.12.01.
[16] Mohammad Zavvar, Meysam Rezaei, Shole Garavand,"Email Spam Detection Using Combination of Particle Swarm Optimization and Artificial Neural Network and Support Vector Machine", International Journal of Modern Education and Computer Science(IJMECS), Vol.8, No.7, pp.68-74, 2016.DOI: 10.5815/ijmecs.2016.07.08.
[17] Yaser Ghaderipour, Hamed Dinari. "A Flow-Based Technique to Detect Network Intrusions Using Support Vector Regression (SVR) over Some Distinguished Graph Features ", International Journal of Mathematical Sciences and Computing (IJMSC), Vol.6, No.4, pp.1-11, 2020. DOI: 10.5815/ijMSC.2020.04.01.

Authors' Profiles

Aasha Singh has completed her M.C.A. degree from KNIT Sultanpur in 2011. She is pursuing her Ph.D. from M.U.I.T. Lucknow. She is presently working as faculty member in the department of Computer Science & Engineering at KNIT Sultanpur. Her research areas are Machine Learning, Software Engineering, Data Mining. She has 05 Years of teaching/industry experience.

Dr. Awadhesh Kumar has completed his B.E. degree from G.B. Pant Engineering College, Pauri (Garhwal) in 1999 and M.Tech. in Computer Science from A.K.T.U. Lucknow, U.P. and Ph.D. from M.N.N.I.T. Allahabad, U.P., India. He is presently working as faculty member in the department of Computer Science & Engineering at KNIT Sultanpur, U.P., India since 2000. His teaching and research interests include Computer Networks, Mobile Ad-Hoc Networks, Wireless Sensor Networks, and Machine Learning.

Dr. Ajay Kumar Bharti, Working as Professor, School of Computer Application, Babu Banarasi Das University, Lucknow. He has over 19 years of rich experience in Research, Education and Industry. He worked in numerous premier organizations like Pixellent Solutions, K.N.I.T. Sultanpur, M.I.E.T. Meerut, University of Lucknow, Lucknow, I.E.T. Lucknow and Maharishi University of Information Technology, Lucknow. His research interest is in Service Oriented Architecture, Knowledge Based System, e-Governance and Artificial Intelligence.

Dr. Vaishali Singh is currently working as an Assistant Professor in Department of Computer Science, Maharishi University of Information Technology, Lucknow. She has received her Ph. D Degree in the year 2017 from B.B. Ambedkar University, Lucknow, India. Her research interest includes Information Retrieval and Question Answering Systems. She has published the research papers in International Conferences and Journals.

How to cite this paper: Aasha Singh, Awadhesh Kumar, Ajay Kumar Bharti, Vaishali Singh, "Pure-Octet Extraction based Technique for Identifying Malicious URLs based on IP Address Attributes", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.12, No.6, pp. 25-32, 2022. DOI:10.5815/ijwmt.2022.06.03