Indonesian Journal of Electrical Engineering and Computer Science
Vol. 16, No. 3, December 2019, pp. 1447~1454
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v16.i3.pp1447-1454

A novel approach to big data analysis using deep belief network for the detection of android malware

Uma Narayanan[1], Varghese Paul[2], Shelbi Joseph[3]
[1,3]Division of Information Technology, Cochin University of Science and Technology, India
[2]Department of Computer Science and Engineering, Rajagiri School of Engineering and Technology, India

Article Info

Article history:

Received Jan 18, 2019
Revised Jun 22, 2019
Accepted Jul 28, 2019

Keywords:

An android malware
Big data
Deep belief network
Deep learning
Security

ABSTRACT

Mobile and tablets are rapidly getting the chance to be basic device in the everyday life. Android has been the most well-known versatile working structure. Regardless, inferable from the open thought of Android, amount of malware is concealed in a broad number of kind applications in Android exhibits that really undermine Android security. Deep learning is another domain of AI explore that has expanded extending thought in artificial information. In this examination, we propose to relate the features from the static examination with features from the dynamic examination of Android applications and depict malware using Deep learning systems. What's more, besides distinguishing sensitive customer data sources is fundamental for security protection in portable applications. So we propose a Novel way to deal with overseeing tremendous information examination utilizing Deep learning for the affirmation of Android malware.

Corresponding Author:

Uma Narayanan
Division of Information Technology,
Cochin University of Science and Technology, Kerala, India.
Email: uma@cusat.ac.in

1. INTRODUCTION

Mobile phones are transforming into the crucial figuring stage for adaptable customers as a result of their assistance for a broad assortment of use. It is typical that the amount of PDA shipments will assemble each year with a record improvement of around 16 % [1]. Ten years back, flexible malware was seen as another and implausible hazard. Various mobile phone customers even have seen themselves as impervious to such risks. Snappy forward to 2017 and more than 1.5 million, McAfee Labs have recognized new events of adaptable malware in the essential quarter of the year alone for a total of more than 16 million portable malware scenes. Today, phones are going under extending attack, and no one is insusceptible. A fourth of respondents didn't realize whether they've experienced an attack. About each of the (94 percent) anticipated that the repeat of portable ambushes should augment, and 79 percent perceived that it's ending up progressively difficult to verify mobile phones.

Mobile malware, as its name proposes, is noxious programming that explicitly focuses on the working frameworks on cell phones. There are numerous kinds of mobile malware variations and various techniques for dispersion and contamination. As more clients are consistently moving far from work area working frameworks and favoring cell phones, instead, it wouldn't have been long until programmers exchanged strategies. At this moment, the volume of versatile dangers is an unimportant part of those that objective work areas. In any case, as increasingly touchy and possibly high-esteem assignments are completed on cell phones, mobile security dangers are quick turning into a developing concern. Different security threats are

a) Spyware and Malware: Malware, short for mobile adware, for the most part, discovers its direction onto a cell phone through the establishment of content or program and regularly without the consent of the

client. The motivation behind most types of malware is to gather information from your telephone to spam you with promotions. Most malware variations ordinarily incorporate a component of spyware, which collects data about your web utilization and sends it on to an outsider. This information may include insights regarding your area, your passwords, and your contacts. That makes it an issue for you, yet conceivably anyone in your contact.

b) Drive-by Downloads: On the off chance that you open the wrong email or visit a pernicious site, you could turn into the casualty of a type of versatile malware known as the drive-by download. These variations are naturally introduced on your gadget and can release a scope of dangers, including spyware, malware, adware or something substantially more genuine, for example, a bot that can utilize your cell phone to perform odious assignments like sending infections to other individuals inside your association or filtering the system for a path in.
c) Viruses and Trojans: What may appear an authentic application could contain an infection or trojan prepared to assault your cell phone. These infections may have a genuinely harmless payload, for example, changing your telephone's backdrop or changing the language. Be that as it may, most have something significantly more harmful at the top of the priority list like digging for passwords and banking data.
d) Mobile Phishing: Phishing endeavors are the same old thing, yet the presentation of the cell phone has seen cybercriminals change their phishing strategies to trick clients of cell phones. Customary phishing strategies include culprits sending messages to clients that seem to start from a trusted source. Mobile phishing makes this strategy one step further and utilizes applications to convey mobile malware. The client, regularly unfit to differentiate between an authentic application and a fake application is unaware as the fake application gathers record numbers, passwords, and the sky is the limit from there.
e) Browser Exploits: With regards to security, your mobile program isn't perfect. Therefore, various program misuses in the wild can exploit your browser and different applications that work inside the program, for example, PDF readers.

The quickly developing examples bring an enormous number of requests for malware recognition in Mobile. With such a significant amount of refined malware tests, a lot of looks into have been focused on proposing various malware identification strategies to relieve the fast development of malware. Malware discovery can be separated into two principle techniques: static malware location and dynamic malware recognition [2, 3]. Static malware identification additionally refers to signature-based malware location, which analyzes the content of malicious binary without really executing malware tests. Signature-based malware detection to get full execution. Be that as it may, it very well may be effectively avoided by obfuscation techniques. What's more, signature-based malware location requires earlier information about malware tests. In light of the limitation of signature-based malware detection, different dynamic malware location techniques have been advanced [4]. Dynamic malware detection examines the example practices amid execution and for the most part, called behavior-based malware identification. Behavior-based malware recognition strategies incorporate virtual machine and capacity call observing, data stream following, and dynamic binary instrumentation.

A noteworthy test is that most malware is infused in generally benign projects. For one specific program, its execution runs could be either benign or harmful relying upon whether the malware introduced is activated, making it hard to mark the program in training. The classifier is additionally very delicate to the decision of generous and malevolent precedents in the preparation set and may result in unsuitable false positives and false negatives. In this paper, we propose an alternate malware recognition situation. Rather than looking for a solitary model that separates all malevolent and benevolent applications, we learn one model for every application which isolates its malware contaminated executions from genuine executions. The model is prepared on both malicious and benign behavior of the application. At the point when the program is stacked, its related conduct model is stacked, and its execution is observed. In the event that the procedure executes in a way that makes the related model banner its conduct as suspicious, a product exemption is raised. Instead of grouping the program as a malignant/amiable as in the "customary" situation, our finder recognizes runs where the contamination is activated from ones where it isn't and raises an exemption when noxious conduct is distinguished.

Early detection of Malware dramatically increases the chances of taking the right decision on a successful plan for the undoing the effect of Malware instead of losing the critical information. Computer systems are applied widely in the detection and differential diagnosis of many different kinds of abnormalities. Therefore, improving the accuracy of a system has become one of the major research areas. In this paper, a new scheme for detection of malware has been developed using deep belief network which uses unsupervised learning method followed by a back- propagation network which uses supervised learning. The construction is a back-propagation neural network learning function while weights are initialized from the deep belief network path (DBN-NN).

Neural systems are a subset of machine learning. They are AI frameworks dependent on recreating associated "neural units," freely displaying how that neuron collaborate in the mind. Computational models enlivened by neural associations have been examined since the 1940s and have come back to noticeable quality as computer processing power has increased and massive training data set have been utilized to effectively dissect input information, for example, pictures, video, and speech. Human-made intelligence experts refer to these strategies as "deep learning," since neural systems have some ("deep") layers of simulated interconnected neurons.

Machine Learning, for example, Decision Tree (DT), K-Nearest Neighbor (KNN), Na¨ıve Bayes (NB), and Support Vector Machine (SVM) are regularly utilized in malware discovery [5, 6]. The customary machine learning calculations can conceivably take in the behavior highlights from the malware. Shockingly, most machine learning calculations' exhibition relies upon the exactness of the behavior highlights. Furthermore, it is frequently hard to remove significant behavior highlights for improving malware discovery execution. Additionally, include preparing requires ability. Like this, conventional machine learning calculations are still to some degree unsatisfying for malware recognition. Deep learning is a part of machine learning that endeavors to gain abnormal state include straightforwardly from the first information. So, Deep learning advocates the start to finish arrangement straightforwardly. It kills the entire procedure of enormous and testing undertaking stage. Deep learning is effective to think about abnormal state highlights of tests by methods for multilayer deep architecture, and it has been broadly utilized in image processing, visual recognition, object detection, and so on [7-12]. Table 1 showing the application of deep learning in a different field.

Table 1. Application of deep learning in different field

Sl. No.	Author	Neural Network	Application
1.	Tushar Anthwal et al. [13]	Back Propagation	Estimation of Trophic Status Index of Lakes
2.	Honglak Lee et al. [14]	Convolutional Deep Belief Networks	For Audio Classification
3.	Fajar Yumono et al. [15]	Back Propagation	For Healthy Chicken Meat Identification
4.	P. Santhi Priya et al. [16]	Multi-Layer Perceptron	Sentiments analysis of Social Media
5.	Jai Utkarsh et.al [17]	Back Propagation	Classification of Atrial Arrhythmias
6.	Arindam Sarkar [18]	Multi-Layer Perceptron	Wireless Communication

2. RESEARCH METHOD

The essential thought of our proposed approach is to distinguish continuously any malware assaults using an all-encompassing and efficient utilization of all conceivable data acquired from the cell phones. In our method, we are using two neural networks Back Propagation and Deep Belief Network. The system architecture is shown in Figure 1.

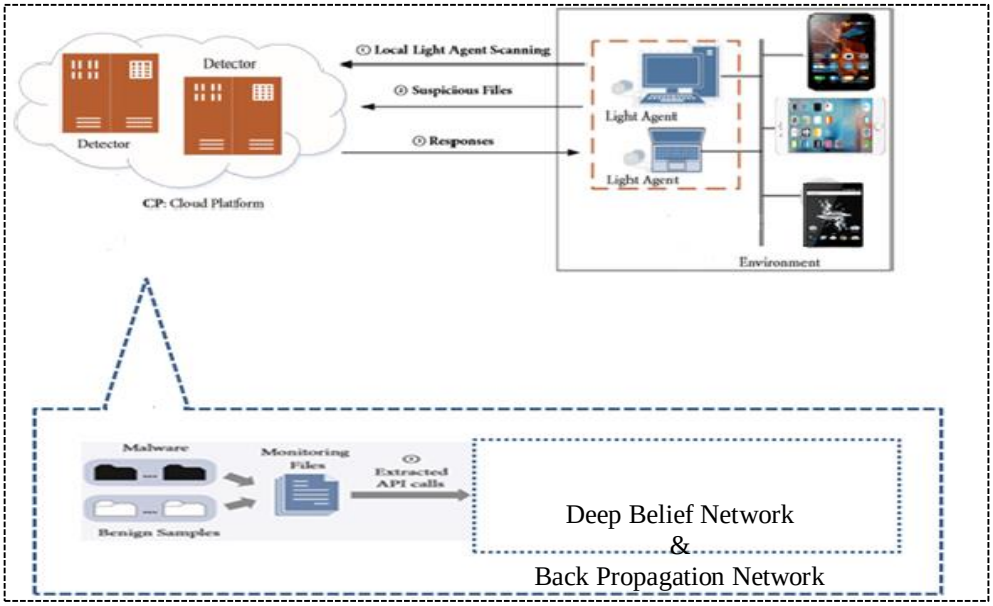

Figure 1. System Architecture

In 1985, the second-generation neural networks with back -propagation algorithm emerged. However, the learning algorithm struggles to adjust network weights so that output neurons state y represent the learning example t. A common method for measuring the discrepancy between the expected output t and the actual output y is using the squared error measure:

$$E = (t - y)^2 \quad (1)$$

The change in weight, which is added to the old weight, is equal to the product of the learning rate and the gradient of the error function, multiplied by -1:

$$\Delta w_{ij} = - \frac{\partial E}{\partial w_{ij}} \quad (2)$$

Back -propagation neural network requires a labeled training data. Therefore, the biggest issue with back propagation Neural Network appear as it's possible to get stuck in poor local optima, and the learning time is huge with multiple hidden layers. Where almost all data is unlabeled then use of Back- propagation is not possible, so we use Back -propagation for fine-tuning. First, we use a Restricted Boltzmann Machine and then Back Propagation. Restricted Boltzmann Machine (RBM) is a generative stochastic artificial neural network that can learn a probability distribution over its set of inputs. On the other hand, a DBN is a generative graphical model, or alternatively a type of deep neural network, composed of multiple layers of latent variables ("hidden units"), with connections between the layers but not between units within each layer.

The Deep Belief Network (DBN) utilizes unlabelled android application tests which uses the unsupervised learning technique to preparing. Boltzmann Machine for which the diagram is making out of two layers, obvious layer, and concealed layer. There is no intra-layer association. DBN uses RBM for planning. Boltzmann Machine is an intermittent neural system with stochastic paired units and undirected edges between units. In light of Boltzmann Machine obstruction to scaling RBM was exhibited, which contains concealed layers unit having restricted relationship between each hidden units. This structure causes RBM to adapt effectively. As DBN is a multi-layer belief network, where each layer is Restricted Boltzmann Machine stacked against one another to frame the Deep Belief Network. The underlying advance of planning DBN is to take in a layer of features from the visible units, using Contrastive Divergence (CD) calculation. Then, the next step is to treat the activations of previously trained features as visible unites and learn features of features in a second hidden layer. At last, the entire DBN was prepared when the learning for the last hidden layer is accomplished. After the proper training of data, it is given to Back Propagation Network with labeled android app data samples for further processing. Figure 2 shows the block diagram of Deep Belief Network with Back Propagation

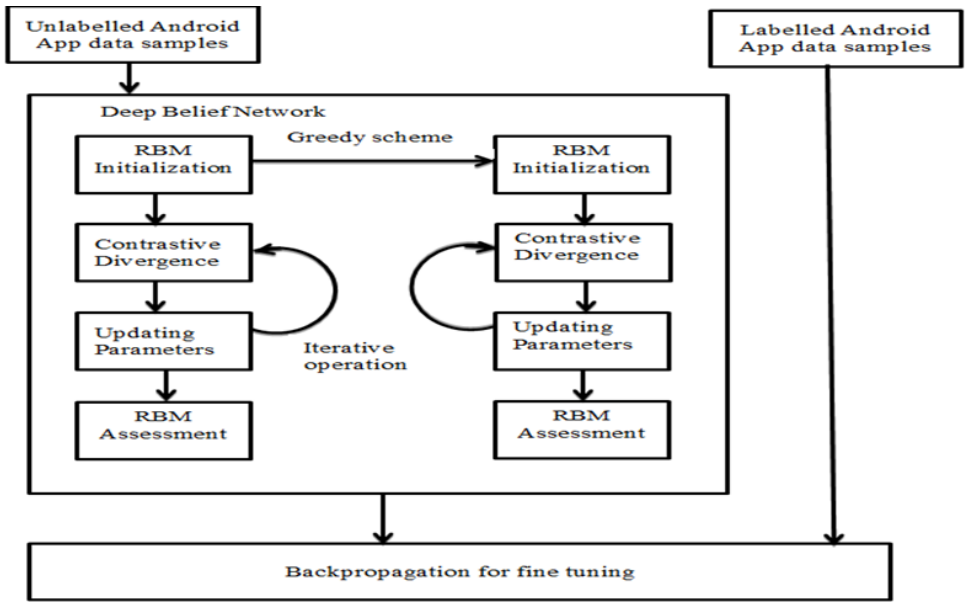

Figure 2. Deep Belief Network and Back Propagation Network

To create information from a RBM, we can begin with an arbitrary state in one of the layers and after that perform substituting Gibbs sampling. The majority of the units in a single layer are refreshed in parallel given the present conditions of the units in the other layer, and this is repeated until the framework is inspecting from its equilibrium. To perform the most extreme probability learning in a RBM, we can utilize the correlation between two relationships. For each weight, wij, between a visible unit i and a hidden unit, j, we measure the connection $<v_i^0\ h_j^0>$ when an information vector is clamped on the visible units, and the hidden states are sampled from their conditional distribution, which is factorial, until it achieves its stationary distribution and measure the correlation $<v_i^\infty\ \ h_j^\infty>$. The gradient of the log likelihood of the preparation information is at that point is

$$\partial \log p(v^0) / \partial w_{ij} = <v_i^0\ h_j^0> - <v_i^\infty\ h_j^\infty> \quad (3)$$

Algorithm 1: Deep Learning model for Malware Detection
Input: Δ including malware and benign sample (S_1, S_2, S_3,….., S_r) sample P_j under detection
Output: C_{out}
(1) Begin
Construct feature vectors S_{jGn} = (v_{j1}, v_{j2}, v_{j3},……, v_{jn})
(2) Activation ={ S_{1Gn}, S_{2Gn}, S_{3Gn}, ……., S_{rGn}}
(3) For k=1 to h do
(4) Train AE[k] use the activation AE[k-1] as the input and train kth hidden layer's parameters
(5) Adjust the weight
(6) End
(7) For i=1 to m do
(8) Initialize weight from above network
(9) Train the Back propagation network
(10) Fine tune the neural network
(11) End
(12) Output the class label C_{out}
(13) End

Contrastive dissimilarity learning in a restricted Boltzmann machine is efficient enough to be practical but when connected in the obvious manner, contrastive divergence learning falls flat for deep, multilayer systems with various loads at each layer in light of the fact that these systems take awfully long even to achieve conditional equilibrium with a clamped data vector. We presently demonstrate that the equality among RBMs and infinite directed nets with tied loads proposes an efficient learning calculation for multilayer organizes in which the loads are not tied. An efficient approach to gain proficiency with a confounded model is to join a lot of easier models that are found out consecutively. To compel each model in the grouping to take in something else from the past models, the information are modified somehow or another after each model has been trained. The thought behind our greedy algorithm is to enable each model in succession to get an alternate representation of data. The model plays out a nonlinear change on its input vectors and delivers as yield the vectors that will be utilized as a contribution for the next model in succession. Algorithm of model is shown in Algorithm 1 in which deep learning is used for the detection of Malware Detection in Mobile phones and Tablets. Table 2 and 3 shows the sample data for training data and the Application details.

Table 2. Example of the Training set

Port number	Port category	Port category value	Class
24	Normal_port	1	0
80	Normal_port	1	0
8080	Normal_port	1	0
6566	Malware_port	2	1
7787	Malware_port	2	1
1090	Malware_port	2	1

Table 3. Application Training set

Application Category	Application Category value	User ID	User ID value	Port Number	Class
Webapp	1	3000	1	80	0
Sysutil	2	0	0	80	0
Unknown	3	0	0	6566	1
Unknown	3	0	0	7787	1

This hybrid model has some attractive features:

1. There is a quick, greedy learning algorithm that can find a genuinely decent arrangement of parameters immediately, even in deep neural systems with a large number of parameters and many hidden layers.
2. The learning calculation is unsupervised; however can be applied to labeled data by learning a model that produces both the label and the data.
3. There is a fine-tuning calculation that learns a superb generative model that outranks discriminative strategies on our database
4. The generative model makes it simple to interpret the distributed representation in the deep hidden layers.
5. The derivation required for framing a percept is both quick and accurate.
6. The learning calculation is local. Changes in accordance with a neurotransmitter quality rely upon just the conditions of the presynaptic and postsynaptic neuron.
7. The communication is basic. Neurons need to convey their stochastic binary states.

3. CONCLUSION

In this research, we displayed a programmed framework for recognizing versatile malware dependent on DBN unsupervised pre-preparing stage pursued by a regulated back proliferation neural system stage (DBN-BPN). The pre-prepared back propagation neural system with unsupervised stage DBN accomplishes higher characterization precision in contrast with a classifier with the only one supervised stage. The method of reasoning behind this improvement could be that the taking in of information measurements from input feature space by DBN stage in states back spread neural system to look target capacity almost a decent neighborhood optima in the supervised learning stage. Our discoveries feature the considerable capability of applying deep learning methods to utilize cases over the economy. The Table 4 shows the advantage of our method. However, we additionally observe some proceeding with confinements and obstructions—alongside future open doors as the advances proceed with their development. At last, the estimation of AI isn't to be found in the models themselves, however in organizations' capacities to saddle them. It is critical to feature that, even as we see financial potential in the utilization of AI strategies, the utilization of information should dependably consider concerns including information security, protection, and potential issues of predisposition.

Table 4. Comparison of Security Approaches

Approach	Working Characteristics	Strength	Limitation
Beehive [19]	Use of PCA and clustering for attack detection. Clustering based correlation.	Identifying previously unknown attacks.	Post-factum threat detection
BotCloud [20]	Use of Page Rank algorithm for C & C botnet detection. Graph-based correlation.	Identifying botnets and their connections	Limited monitoring scope (i.e., network logs).
Large-scale botnet detection [21]	Use of common packet characteristics. Clustering based correlation.	Detecting well-known botnet attacks	Limited against mimicry/ hidden attacks
Our Approach	Classifiers at individual and aggregate levels, for determination of attack presence.	Real-time threat detection. Detecting mimicry/ hidden attacks. Wide monitoring scope.	Occasional latency

ACKNOWLEDGEMENTS

I respect and thank Prof. Dr.Varghese Paul, for providing me an opportunity to do the project work in CUSAT and giving me all support and guidance, which made me complete the project duly. I am extremely thankful for providing such friendly support and guidance, although he had a busy schedule.

REFERENCES

[1] "Developer Works survey," 2013. Available: http://public.dhe.ibm.com/software/dw/survey/2010surveyresults/2010surveresults-pdf.pdf.

[2] S. Cesare, et al., "Control flow-based malware variant detection," *IEEE Transactionson Dependable and Secure Computing*, vol. 11, pp. 304-317, 2014.

[3] H. S. Galal, et al., "Behavior-based features model for malware detection," *Journal in Computer Virologyand Hacking Techniques*, vol. 12, pp. 59-67, 2016.

[4] A. Kharraz, et al., "UNVEIL: a large-scale, automated approach to detecting ransomware," *Proceedings of the USENIX Security Symposium*, pp. 757-772, 2016.

[5] M. Fan, et al., "Android malware familial classification and representative sample selection via frequent subgraph analysis," *IEEE Transactionson Information Forensics and Security*, vol. 13, pp. 1890-1905, 2018.

[6] Z. Lin, et al., "Asecure encryption-based malware detection system," *KSII Transactions on Internet and Information Systems*, vol. 12, pp. 1799-1818, 2018.

[7] H. William, et al., "DL4MD: A deep learning framework for intelligent malware detection," *Proceedings of the International Conference on Data Mining (DMIN), The Steering Committee of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp)*, pp. 61, 2016.

[8] O. Ronneberger, et al., "U-net: convolutional networks for biomedical image segmentation," *Proceedings of the International Conference on Medical Image Computing Mathematical Problems in Engineering and Computer-Assisted Intervention (MICCAI '15), of Lecture Notes in Computer Science*, Springer, Cham, Switzerland, vol. 9351, pp. 234-241, 2015.

[9] W. Yang, et al., "Down image recognition based on deep convolutional neural network," *Information Processing in Agriculture*, vol. 5, pp. 246-252, 2018.

[10] J. Donahue, et al., "Long-term recurrent convolutional networks for visual recognition and description," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 2625-2634, 2015.

[11] A. Voulodimos, et al., "Deep learning for computer vision: a brief review," *Computational Intelligence and Neuroscience*, vol. 2018, 2018.

[12] S. Ren, et al., "Faster R-CNN: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, pp. 91-99, 2015.

[13] T. Anthwal, et al., "Performance Analysis of ANN Model for Estimation of Trophic Status Index of Lakes," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 7, pp. 1-10, 2018.

[14] H. Lee, et al., "Unsupervised feature learning for audio classification using convolutional deep belief networks," *Proceedings of the 22nd International Conference on Neural Information Processing System*, Vancouver, Canada, pp. 1096-1104, 2009.

[15] F. Yumono, et al., "Artificial Neural Network for Healthy Chicken Meat Identification," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 7, pp. 63-70, 2018.

[16] P. S. Priya and T. V. S. Rao, "Analysing Event-Related Sentiments on Social Media with Neural Networks," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 7, pp. 119-124, 2018.

[17] J. Utkarsh, et al., "Classification of Atrial Arrhythmias using Neural Networks," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 7, pp. 90-94, 2018.

[18] A. Sarkar, "Multilayer neural network synchronized secured session key based encryption in wireless communication," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 8, pp. 44-53, 2019.

[19] T. F. Yen, et al., "Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks," *Proc. 29th Annu. Comput. Secur. Appl. Conf.*, pp. 199-208, 2013.

[20] J. Francois, et al., "BotCloud: Detecting botnets using MapReduce," *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, pp. 1-6, 2011.

[21] K. Singh, et al., "Big data analytics framework for peer-to-peer botnet detection using random forests," *Inf. Sci.*, vol. 278, pp. 488-497, 2014.

BIOGRAPHIES OF AUTHORS

Uma Narayanan is a research scholar in the Division of Information Technology, Cochin University of Science and Technology, Kerala, India. She received a bachelor's degree in Computer Science and Engineering and master's degree in Network Engineering from Mahatma Gandhi University, Kerala, India. She is master in information security, and mainly engages in big data security research.

Varghese Paul received his B.Sc (Engg) in Electrical Engineering from Kerala University, M.Tech in Electronics and Ph.D in Computer Science from Cochin University of Science and Technology. Research Supervisor of Cochin University of Science and Technology, M G University Kottayam, Anna Technical University Chennai, Bharathiar University Coimbatore, Bharathidasan University Trichy and Karpagam University Coimbatore. Under the guidance, 29 research scholars had already completed research studies and degree awarded. Research areas are Data Security using Cryptography, Data Compression, Data Mining, Image Processing and E_Governance. Developed TDMRC Coding System for character representation in computer systems and encryption system using this unique coding system. Published many research papers in international as well as national journals and a text book also.

Shelbi Joseph received the BE. Degree from University of Madras in 1992 in Computer Science and M.Tech degree in Computer Science from Department of Computer Science, National Institute of Technology, Tiruchirappalli in 2006. He spent seven years in software industry, and currently working as Assistant professor, Division of Information Technology, School of Engineering, Cochin University of Science and Technology. He carried out his research work leading to Ph.D at School of Engineering, Cochin University of Science and Technology in Software Reliability. His areas of interest are Software Engineering, Software Reliability, Open Source Software , Big Data ,Data Mining and IOT. He has number of publications in National and International Journals and Conference proceedings to his credit.