International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-10 Issue-11, September 2021

Mitigation of Malware Effect using Cyber Threat Analysis using Ensemble Deep Belief Networks

K. Janani

Abstract: ***Cybersecurity is a technique that entails security models development techniques to the illegal access, modification, or destruction of computing resources, networks, program, and data. Due to tremendous developments in information and communication technologies, new dangers to cyber security have arisen and are rapidly changing. The creation of a Deep Learning system requires a substantial number of input samples and it can take a great deal of time and resources to gather and process the samples. Building and maintaining the basic system requires a huge number of resources, including memory, data and computational power. In this paper, we develop an Ensemble Deep Belief Networks to classify the cybersecurity threats in large scale network. An extensive simulation is conducted to test the efficacy of model under different security attacks. The results show that the proposed method achieves higher level of security than the other methods.***

Keywords: Cybersecurity, Deep Learning, Ensemble Deep Belief Network, Attacks.

I. INTRODUCTION

In order to ensure confidentiality, integrity, and availability of computer resources, networks, software programmes, and attack data, cyber security is used to collect policies, techniques, technologies, and processes that function together [1]. However, many opponents still benefit from the fact that only one weakness in systems that need security needs to be found [2]. With increasing numbers of Internet-connecting systems, the attack surface also grows and the risk of attacks is higher [3]. In addition, assailants are growing smarter, producing no-day exploits and spyware avoids security protection and enable them to continue without detection for lengthy durations [4]. Zero-day misuses are assaults that were not before attacked, but typically vary from a known attack [5]. To compound the problem, attack methodologies are commoditized, which allows fast distribution without requiring the user to know how to construct exploits. Defenders must also guard against insider risks from people or organisations who misuse their permitted access, as well as against external threats [6]. There is evidence of compromise throughout the life cycle of an attack; warnings of an imminent attack may even exist. The difficulty is finding these signs that can be spread throughout the environment [7]. There is a vast amounts of data rendered by human-to-machine and machine-to-machine interactions from servers, applications, and smart devices resources [8]. Cyber defence systems [19]-[23] generate large amounts of data, such as the SIEM system, which constantly overloads the guard analyst with event warnings. The use of cyber security can coordinate events, discover trends and detect anomalies in the behaviour of a defensive programme to improve its security position. The creation of cyber-defense systems using data analytics is starting to occur [8]. In this paper, we develop an Ensemble Deep Belief Networks to classify the cybersecurity threats in large scale network. An extensive simulation is conducted to test the efficacy of model under different security attacks.

II. RELATED WORKS

Many approaches to detecting malware are available. The second study improves on the first, employing DL [9] [10] detectors of malicious Android applications with functionality from static and dynamical analyses. The characteristics were derived specifically from three sources: the static analysis of necessary permissions, sensitive APIs and dynamic behaviour. This gives the necessary permissions and uses the APIs. The dynamic features come from dynamic analysis by collecting data from the Android sandbox, DroidBox. Several configurations have been tested and the most successful was the two-hidden layer DBN.

Dynamic traits are more dependable than static characteristics, which can readily be obscured. Pascanu et al. [11] have developed a method for malware detection that employs mixed logistic regression and multiple-layer classification perceptron RNNs. CNNs and RNNs have been used to identify malware by Kolosnjaji et al. [12]. The list of sequences to call the API kernel will be transformed by one-hot encoding into binary vectors. One-hot coding is an easy-to-use encoding approach to store categorical data. This data is used to train the CNN and RNN DL algorithms.

Tobiyama et al. [13] have created a malware detector that supplies the API to the RNN to extract time series data. These characteristics are transformed into images, and they are classified as malicious or normal by CNN. The RNN uses LSTM, while the CNN has two overlapping layers and two layers of pooling. Two completely connected layers follow this. Although the data set was rather modest, an AUC of 0.96 was achieved.

By pre-processing the Windows Portable Executable (PE) files, Ding et al. [14] constructed a DBN for the extraction of the n-grams.

Manuscript received on September 02, 2021.
Revised Manuscript received on September 08, 2021.
Manuscript published on September 30, 2021.
* Correspondence Author
K. Janani*, Research Scholar, Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore (Tamil Nadu), India.

Three layers of the DBN were hidden. When prepared with unlabeled data, the DBN model outperforms decision trees, SVMs, and K-NN. McLaughlin et al. [15] also employed malware file opcodes to create a detector without the selection of any feature or technology. This method employed a raw opcode processing layer to fill in a CNN with a two-layer convolution, max-pooling, and fully linked layer and a layer of classification. The large fall in the first and second datasets is probably due to a considerable increase in the malware variability of the second dataset and corresponds to the decrease in non-DL approaches. Saxe and Berlin [16] used a Bayesian calibration model that gives a chance of malware being a file. This is based on a previous malware ratio and the error rate of the DSN, utilising an Epanechikov kernel to estimate the kernel density, as a standard distribution cannot be assumed. Shibahara et al. [17] offered the approach of deciding if network-based, dynamical analysis of network data should be done, especially when malware stops C2 activity, and if it should be discontinued by network behaviour. For that reason, the repetitive neural tensor network employed enabled high classification performance to be calculated by employing a tensor to improve the performance. Chen et al. [18] trained DBN in an unmonitored way, followed by an inventory layer, with a hidden layer of RBM. The approach is of a softmax, decision-making trees, SVMs, and random woodland with accuracies between 91% and 96%, depending on the ratio between normal and malevolent. But without additional measures, such as real positive and false positive rates, these results are difficult to understand.

III. PROPOSED METHOD

In this section, the ensemble of deep belief network is composed for the process of classifying the cyber security threats. These ensemble models are a mixture of Restricted Boltzmann machine and Deep Auto Encoders of Deep Belief Networks. Here, the Restricted Boltzmann machine acts as the base classifier and Deep Auto Encoders acts as the meta classifiers.

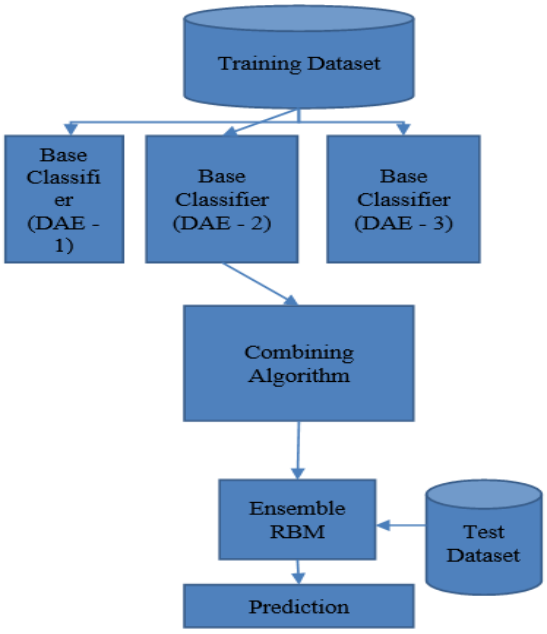

Figure 1: Proposed Ensemble Detection or Classification Model

A. Deep Belief Network Classification

Deep Belief Networks (DBNs) are the Deep Neural Networks class that are built of several layers of hidden units that contain layer-specific connections but do not contain units inside each layer. DBNs are uncontrollably trained. They are often trained to independently rebuild the inputs by altering weights in every hidden layer.

B. Deep Autoencoders

Autoencoders are a class of uncontrolled neural networks where the network uses a vector for input and attempts to match the output with that same vector. Input can be taken and the dimension can be changed. The data can be represented in greater or lesser dimensions. These kinds of neural networks are unbelievably adaptable as they learn unattended compressed data encoding. In addition, the computer resources needed to develop an effective model can be trained at a single layer.

The network is utilised to encode the data when the hidden levels have fewer dimensions than the input and output layers (Fig. 2). A noise-driving autoencoder can be developed to be more resilient through the training of an automatic autoencoder to restructure the input of a version of the input. This approach has proven to be more generalised and more resilient than standard autoencoders.

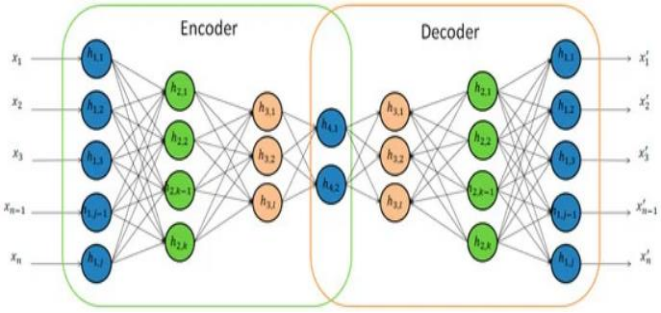

Figure 2. Deep autoencoder.

The use of many layers of series trained auto-encoders is called stacked auto-encoders to progressively compress the information (Figure 3). The full stacked categorization layer autoencoder is in Figure 3, followed by the auto encoder and then the outputs. These are joined and a classification layer is added once they are trained. As with ordinary autoencoders, autoencoders that denoise can also be stacked.

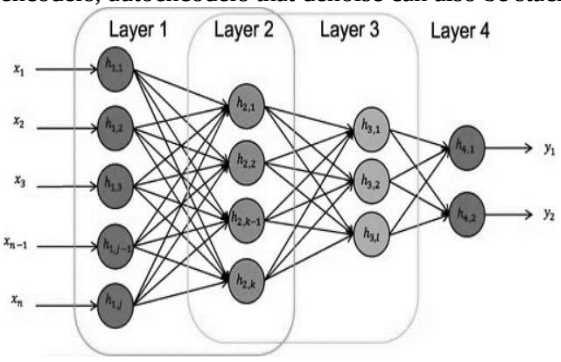

Figure 3. Stacked autoencoder with a classification layer

The sparse auto-encoder is a sort of encoder in which hidden nodes exist, but only a portion of the hidden units are activated at a given time.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

This means that additional nodes are penalised.

C. Restricted Boltzmann Machines

The Restricted Boltzmann Machines (RBMs) are two, bipartisan, undirected visual models which compose the DBN building blocks, and not a single one. RBMs are unattended and can be trained on a layer at a time, similar to auto-encoders. The input layer is the first layer and the hidden layer is the second layer (Figure 4). No intra-layer connections are available, but, every input layer node in the hidden layer is connected to every other node.

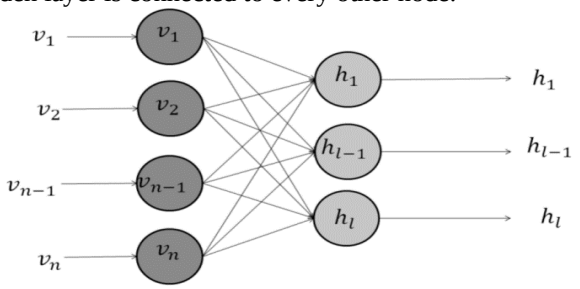

Figure 4. Restricted Boltzmann machine

In general, binary units are limited to input and hidden layers. The network is trained to reduce "energy," which assesses the compatibility of the model and uses statistical mechanics to do a great deal of mathematics. The purpose of model training is to determine the functions that reduce the energy of the system and, consequently, the hidden state. RBMs are also probabilistic, i.e., instead of explicit values, they assign probability levels. The output can be utilised as functionality for a different model. The model is trained through the transfer and forwarding of binary input data via the model. Then the input data will be re-enabled by the model. The system energy is then calculated and employed for weight upgrading. It continues until the model converges. RBMs can likewise be layered into several layers in order to produce a deeper neural network for autoencoders, which is the stacked RBM.

D. Classification Layers of Ensemble DBNS

The classification layers (Figure 5) may be used for the classification of both RBM and autoencoders with completely connected layers or layers. Uncontrolled learning layers are employed as functional extractors and form inputs into fully connected layers that are trained via back propagation. For the purpose of classification.

The input data $S = (S_1, S_2..., S_i,\ldots,S_n)$ is sent across the layers $v = (v_1, v_2..., v_n)$. The classification is conducted in the hidden layer h where, $h = (h_1, h_2..., h_m)$ that helps in classification via learning the features F from the input base classifiers.

The classification is taken as the event set (v,h) with $E(v,h|\theta)$ being the energy function for training the DBN,

$$E\left(v,h|\theta\right)=-\sum_{i=1}^{n}a_i v_i-\sum_{j=1}^{m}b_i h_i-\sum_{i=1}^{n}\sum_{j=1}^{n}v_i w_{ij} h_j$$

where in Eq.(1),

θ-$\{W_{ij}, a_i, b_j\}$

w is regarded as the connection of weights between the layers, and
a is regarded as the visible layer bias and
b is regarded as the hidden layer bias.
The probability distribution is hence modelled as below:

$$P\left(v,h|\theta\right)=\frac{e^{-E\left(v,h|\theta\right)}}{Z\left(\theta\right)}$$

$$Z\left(\theta\right)=\sum_{v,h}e^{-E\left(v,h|\theta\right)}$$

Similarly, the conditional probability distribution is hence represented as below for two different sampling process say a and b:

$$P\left(h_j=1|v,\theta\right)=sigmoid\left(b_j+\sum_i v_i w_{ij}\right)$$

$$P\left(v_j=1|v,\theta\right)=sigmoid\left(a_j+\sum_i h_i w_{ji}\right)$$

This helps to determine the activation state of each layer that determines the process of classification.

IV. RESULTS AND DISCUSSIONS

In this section, we present the classification accuracy of various methods i.e. proposed ensemble DBN and existing deep learning classifiers. The proposed research work is compared with existing methods in terms of accuracy, recall, precision and F1-score. The results of evolution is conducted against five different datasets that includes Alexa Top Sites, OSINT, DGArchive, Google Play Store and VirusTotal service. For the purpose of testing, the study considers as in the first column of Table 1 that includes Contagio, Genome Project and VirusTotal.

Table 1: Results of Recall against various malwares present in the dataset

Malware Type Selected	Classifiers	Dataset				
		Alexa Top Sites	OSINT	DG Archive	Google Play Store	Virus Total service
Contagio	DNN	0.8315	0.7950	0.6670	0.6305	0.4660
	SAE	0.8498	0.8406	0.7401	0.6944	0.5848

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

	DAE	0.8955	0.8589	0.8498	0.7675	0.6579
	DBN	0.9046	0.9046	0.8955	0.8863	0.8955
	Ensemble DBN	0.9915	0.9915	0.9915	0.9815	0.9915
Genome Project	DNN	0.8406	0.7401	0.6213	0.5026	0.4477
	SAE	0.8589	0.8498	0.8224	0.7438	0.5391
	DAE	0.8680	0.8772	0.7493	0.6880	0.6579
	DBN	0.8955	0.9046	0.9046	0.9000	0.9046
	Ensemble DBN	0.9915	0.9915	0.9915	0.9935	0.9915
Virus Total	DNN	0.8315	0.7584	0.6944	0.6305	0.4843
	SAE	0.9046	0.8955	0.8955	0.9055	0.9046
	DAE	0.9046	0.8955	0.9046	0.8726	0.8772
	DBN	0.8772	0.8772	0.7584	0.6606	0.6488
	Ensemble DBN	0.9414	0.9314	0.9214	0.8854	0.6911

From the Table 1, it is seen that the proposed method has improved recall rate than other existing methods. It is interpreted that the proposed ensemble model performs well on classification than other existing models over different malwares present in the dataset.

Table 2: Results of Precision against various malwares present in the dataset

Malware Type Selected	**Classifiers**	**Dataset**				
		Alexa Top Sites	**OSINT**	**DG Archive**	**Google Play Store**	**Virus Total service**
Contagio	DNN	0.8406	0.7584	0.6579	0.5665	0.5757
	SAE	0.8863	0.8589	0.8315	0.7493	0.7493
	DAE	0.8132	0.7858	0.7767	0.7858	0.7858
	DBN	0.8955	0.8772	0.8955	0.8863	0.8955
	Ensemble DBN	0.9138	0.9138	0.9138	0.9046	0.9138
Genome Project	DNN	0.8132	0.7401	0.6579	0.5501	0.4934
	SAE	0.8863	0.8589	0.8315	0.7493	0.7127
	DAE	0.7858	0.8041	0.7767	0.7858	0.7858
	DBN	0.8861	0.8861	0.8861	0.8981	0.9046

	Ensemble DBN	0.9046	0.8955	0.9046	0.8973	0.8955
Virus Total	DNN	0.7950	0.7401	0.6213	0.5482	0.4934
	SAE	0.8772	0.8589	0.8406	0.7401	0.7127
	DAE	0.7938	0.7846	0.7938	0.7708	0.7661
	DBN	0.8772	0.8772	0.8772	0.8964	0.8955
	Ensemble DBN	0.9018	0.8927	0.9018	0.8982	0.9018

From the Table 2, it is seen that the propose research work has improved precision rate than other existing methods. It is interpreted that the proposed ensemble model performs well on classification than other existing models over different malwares present in the dataset.

Table 3: Results of Accuracy against various malwares present in the dataset

Malware Type Selected	Classifiers	Dataset				
		Alexa Top Sites	OSINT	DG Archive	Google Play Store	Virus Total service
Contagio	DNN	0.8589	0.8041	0.6853	0.6213	0.4751
	SAE	0.9046	0.9046	0.9046	0.9046	0.9046
	DAE	0.9046	0.9046	0.9046	0.8955	0.8863
	DBN	0.9046	0.8680	0.7493	0.6944	0.6670
	Ensemble DBN	0.8769	0.8400	0.8308	0.7938	0.6554
Genome Project	DNN	0.7950	0.6762	0.6305	0.5117	0.4203
	SAE	0.9046	0.9046	0.9046	0.9046	0.8955
	DAE	0.9046	0.9046	0.9046	0.8955	0.8955
	DBN	0.9138	0.8677	0.7569	0.6738	0.6738
	Ensemble DBN	0.8315	0.8224	0.8315	0.7584	0.5482
VirusTotal	DNN	0.8041	0.7127	0.6031	0.4843	0.4203
	SAE	0.9046	0.9046	0.9046	0.9046	0.9046
	DAE	0.9138	0.9046	0.9046	0.8954	0.8769
	DBN	0.8589	0.8406	0.7401	0.6762	0.6670
	Ensemble DBN	0.8472	0.8290	0.8290	0.7470	0.6559

From the Table 3, it is seen that the proposed research work has improved accuracy rate than other existing methods. It is interpreted that the proposed ensemble model performs well on classification than other existing models over different malwares present in the dataset.

Table 4: Results of F1-measure against various malwares present in the dataset

Malware Type Selected	Classifiers	Dataset				
		Alexa Top Sites	OSINT	DG Archive	Google Play Store	Virus Total service
Contagio	DNN	0.8406	0.7584	0.6762	0.5848	0.5117
	SAE	0.9046	0.8772	0.8406	0.7584	0.7219
	DAE	0.8041	0.7950	0.7858	0.7950	0.7493
	DBN	0.8955	0.8955	0.8863	0.8863	0.8863
	Ensemble DBN	0.9138	0.9138	0.9138	0.9138	0.9046
Genome Project	DNN	0.8315	0.7401	0.6579	0.5665	0.4660
	SAE	0.8680	0.8498	0.7950	0.7401	0.7127
	DAE	0.7767	0.7675	0.7675	0.7675	0.7584
	DBN	0.8861	0.8769	0.8861	0.8769	0.8861
	Ensemble DBN	0.9046	0.9046	0.9046	0.9046	0.8955
VirusTotal	DNN	0.8498	0.7493	0.7127	0.6396	0.4934
	SAE	0.8955	0.8680	0.8315	0.7493	0.7219
	DAE	0.8123	0.7938	0.7754	0.7569	0.7661
	DBN	0.8955	0.8772	0.8863	0.8772	0.8772
	Ensemble DBN	0.9018	0.9018	0.9018	0.8927	0.8927

From the Table 4, it is seen that the proposed research work has improved F1-rate than other existing methods. It is interpreted that the proposed ensemble model performs well on classification than other existing models over different malwares present in the dataset.

V. CONCLUSIONS

In this paper, Ensemble Deep Belief Networks is used to classify the cybersecrity attacks in case of large scale IoT networks. The utilisation of Restricted Boltzmann machine and Deep Auto Encoders of Deep Belief Networks enables improved correlation between the elements present in the datasets. The operation of Restricted Boltzmann machine acts as the base classifier and Deep Auto Encoders acts as the meta-classifier improves the accuracy of classification. The results of simulation shows an improved efficacy on test datasets over other cybersecurity attacks. The results show that the proposed method achieves higher degree of attack detection than the other methods.

REFERENCES

1. Buczak, A. L., & Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications surveys & tutorials*, *18*(2), 1153-1176.
2. Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, *28*(1-2), 18-28.
3. Torres, J. M., Comesaña, C. I., & Garcia-Nieto, P. J. (2019). Machine learning techniques applied to cybersecurity. *International Journal of Machine Learning and Cybernetics*, *10*(10), 2823-2836.
4. Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., ... & Wang, C. (2018). Machine learning and deep learning methods for cybersecurity. *Ieee access*, *6*, 35365-35381.
5. Sarikaya, R., Hinton, G. E., & Deoras, A. (2014). Application of deep belief networks for natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *22*(4), 778-784.
6. Larriva-Novo, X., Villagrá, V. A., Vega-Barbas, M., Rivera, D., & Sanz Rodrigo, M. (2021). An IoT-Focused Intrusion Detection System Approach Based on Preprocessing Characterization for Cybersecurity Datasets. *Sensors*, *21*(2), 656.

7. Pérez, S. I., Moral-Rubio, S., & Criado, R. (2021). A new approach to combine multiplex networks and time series attributes: Building intrusion detection systems (IDS) in cybersecurity. *Chaos, Solitons & Fractals*, *150*, 111143.
8. Ustun, T. S., Hussain, S. M., Ulutas, A., Onen, A., Roomi, M. M., & Mashima, D. (2021). Machine Learning-Based Intrusion Detection for Achieving Cybersecurity in Smart Grids Using IEC 61850 GOOSE Messages. *Symmetry*, *13*(5), 826.
9. Yuan, Z., Lu, Y., Wang, Z., & Xue, Y. (2014, August). Droid-sec: deep learning in android malware detection. In *Proceedings of the 2014 ACM conference on SIGCOMM* (pp. 371-372).
10. Yuan, Z., Lu, Y., & Xue, Y. (2016). Droiddetector: android malware characterization and detection using deep learning. *Tsinghua Science and Technology*, *21*(1), 114-123.
11. Pascanu, R., Stokes, J. W., Sanossian, H., Marinescu, M., & Thomas, A. (2015, April). Malware classification with recurrent networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1916-1920). IEEE.
12. Kolosnjaji, B., Zarras, A., Webster, G., & Eckert, C. (2016, December). Deep learning for classification of malware system call sequences. In *Australasian joint conference on artificial intelligence* (pp. 137-149). Springer, Cham.
13. Tobiyama, S., Yamaguchi, Y., Shimada, H., Ikuse, T., & Yagi, T. (2016, June). Malware detection with deep neural network using process behavior. In *2016 IEEE 40th annual computer software and applications conference (COMPSAC)* (Vol. 2, pp. 577-582). IEEE.
14. Ding, Y., Chen, S., & Xu, J. (2016, July). Application of deep belief networks for opcode based malware detection. In *2016 International Joint Conference on Neural Networks (IJCNN)* (pp. 3901-3908). IEEE.
15. McLaughlin, N., Martinez del Rincon, J., Kang, B., Yerima, S., Miller, P., Sezer, S., ... & Joon Ahn, G. (2017, March). Deep android malware detection. In *Proceedings of the seventh ACM on conference on data and application security and privacy* (pp. 301-308).
16. Saxe, J., & Berlin, K. (2015, October). Deep neural network based malware detection using two dimensional binary program features. In *2015 10th International Conference on Malicious and Unwanted Software (MALWARE)* (pp. 11-20). IEEE.
17. Shibahara, T., Yagi, T., Akiyama, M., Chiba, D., & Yada, T. (2016, December). Efficient dynamic malware analysis based on network behavior using deep learning. In *2016 IEEE Global Communications Conference (GLOBECOM)* (pp. 1-7). IEEE.
18. Chen, Y., Zhang, Y., Maharjan, S., Alam, M., & Wu, T. (2019). Deep learning for secure mobile edge computing in cyber-physical transportation systems. *IEEE Network*, *33*(4), 36-41.
19. Raja, R. A., Yuvaraj, N., & Kousik, N. V. (2021). Analyses on Artificial Intelligence Framework to Detect Crime Pattern. Intelligent Data Analytics for Terror Threat Prediction: Architectures, Methodologies, Techniques and Applications, 119-132.
20. Chang, V., Gobinathan, B., Pinagapani, A., Kannan, S., Dhiman, G., & Rajan, A. R. (2021). Automatic detection of cyberbullying using multi-feature based artificial intelligence with deep decision tree classification. Computers & Electrical Engineering, 92, 107186.
21. Karthikeyan, T., Praghash, K., & Reddy, K. H. (2021). Binary Flower Pollination (BFP) Approach to Handle the Dynamic Networking Conditions to Deliver Uninterrupted Connectivity. Wireless Personal Communications, 1-20.
22. Sara, S. B. V., Anand, M., Priscila, S. S., Manikandan, R., & Ramkumar, M. (2021). Design of autonomous production using deep neural network for complex job. Materials Today: Proceedings.
23. Kousik, N. V., Sivaram, M., Yuvaraj, N., & Mahaveerakannan, R. (2021). Improved Density-Based Learning to Cluster for User Web Log in Data Mining. In Inventive Computation and Information Technologies (pp. 813-830). Springer, Singapore.

AUTHOR PROFILE

K. Janani, M.C.A,M.Phil., Research Scholar, Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore, India

Publication:

1. Ransomware Attack, Analysis and Consequences of Malicious File, K.Janani, R.Gunasundari, Karpagam JCS Vol.16 Issue 3 May-June 2021.