Artificial Intelligence Research and Development
A. Cortés et al. (Eds.)

doi:10.3233/FAIA220323

Deep Neural Classification of Darknet Traffic

Mahmoud Alimoradi[a], Mahdieh Zabihimayvan[b,1], Arman Daliri [c], Ryan Sledzik [d], and Reza Sadeghi [e]

[a, c] *Independent researchers*

[b,d] *Department of Computer Science, Central Connecticut State University, New Britain, CT, USA*

[e] *School of Computer Science and Mathematics, Marist College, Poughkeepsie, NY, USA*

Abstract. Darknet is an encrypted portion of the internet for users who intend to hide their identity. Darknet's anonymous nature makes it an effective tool for illegal online activities such as drug trafficking, terrorist activities, and dark marketplaces. Darknet traffic recognition is essential in monitoring and detection of malicious online activities. However, due to the anonymizing strategies used for the darknet to conceal users' identity, traffic recognition is practically challenging. The state-of-the-art recognition systems are empowered by artificial intelligence techniques to segregate the Darknet traffic data. Since they rely on processed features and balancing techniques, these systems suffer from low performance, inability to discover hidden relations in data, and high computational complexity. In this paper, we propose a novel decision support system named Tor-VPN detector to classify raw darknet traffic into four classes of Tor, non-Tor, VPN, and non-VPN. The detector discovers complex non-linear relations from raw darknet traffic by our deep neural network architecture with 79 input artificial neurons and 6 hidden layers. To evaluate the performance of the proposed method, analyses are conducted on a benchmark dataset of DIDarknet. Our model outperforms the state-of-the-art neural network for darknet traffic classification with an accuracy of 96%. These results demonstrate the power of our model in handling darknet traffic without using any preprocessing techniques, like feature extraction or balancing techniques.

Keywords. Darknet traffic, Machine learning, Decision support system, Deep neural network, Tor, Classification.

1. Introduction

Anonymity networks complicate any possibility of tracking and tracing of users' identity on the Web and rely on a worldwide network of volunteer Web servers. Darknets such as Tor and I2P are anonymity networks that prevent traffic analysis and activity monitoring using encryption schemes like onion routing [1]. The anonymity on darknets is indeed provided for both senders and receivers. This anonymous nature allows users to carry on illegal activities as dark hidden services. A web of such services on darknets such as Tor is called dark Web and there has been a great deal of work to analyze the content and application of hidden services on dark Web [2] [3]. However, the focus of this paper is on classification of network traffic on darknets, rather than investigation of dark Web.

[1] Corresponding Author; E-mail: zabihimayvan@ccsu.edu.

Darknet traffic classification plays an important role in detection of cyberattacks and malicious activities on the Internet [4] [5]. There have been significant efforts to detect and classify encrypted traffic of different darknets that rely on machine learning techniques. Hu et. al. propose a hierarchical classification method to identify the type of traffic (darknet or regular internet), type of darknet (Tor, I2P, ZeroNet, Freenet), and user behavior on each network [6]. Choorod and Weir propose a character frequency approach to classify Tor traffic based on characteristics of the encrypted payload. They employ and evaluate different machine learning methods to distinguish Tor packets from regular Web traffic [7]. However, there is few studies on evaluation of deep neural networks to detect and characterize darknet traffic. Perhaps our best understanding of deep neural networks as darknet classifiers in from Lashkari et. al. [8], who proposed a method based on convolutional neural networks to classify darknet traffic. Their method utilizes a feature selection technique to find the most important features and create a gray image that is fed into a two-dimensional convolutional neural network to detect and characterize traffic. Following motivations led to our study:

- The data that is used for darknet traffic classification should be a recent benchmark data that not only can be accessible by other researchers, but also contain both anonymized VPN and Tor activity traffic to represent the real darknet traffic.
- Darknet traffic data contains a large list of features for traffic samples and many related studies employ different feature selection techniques to reduce the number of features to a number that is manageable by existing machine learning methods.
- Highly imbalanced data is naturally inherent in cybersecurity applications such as darknet traffic classification, fraud, and phishing attack detection [9]. This can pose difficulty and inefficiency to machine learning methods due to bias towards majority class. However, balancing data using oversampling (or undersampling) can discard useful information about the data that can be crucial for classification [10].

In this paper, we propose a novel deep neural network to classify traffic data into four classes of Tor, VPN, non-Tor, and non-VPN. The experiments rely on a large dataset that is recently collected and published on Kaggle [8]. We propose a deep neural network as the classifier to distinguish between regular and darknet traffic. Our model can also handle the high-class imbalance without any preprocessing technique to balance the data. The experimental results indicate that out proposed deep neural network outperforms the state-of-the-art deep neural network for darknet traffic classification with the accuracy and F1 of 96%, and Kappa value of 0.92. The neural network we propose in this work can also identify salient features in traffic data with no need for a feature extraction technique prior to detection.

The rest of this paper is organized as follows: in Section 2, we first discuss the related work on darknet traffic characterization and classification. Section 3 provides a background knowledge on deep neural networks and describes the network proposed in this study. Section 4 presents the experiments to evaluate the performance of the proposed model and Section 5 discusses the conclusion and future direction of our work.

2. Related Work

There has been recently a great deal of effort on darknet traffic classification although the emergence of Web traffic classification backs to two decades ago [11-15]. Nishikaze et al. propose a new system based on machine learning techniques to monitor malicious activities on the Internet [16]. The packets studied in their work were captured in a communication from a source network to a dark net. For each packet, 27 categories of traffic analysis profile were created in the form of a 27-dimensional feature vector. They used hierarchical clustering to identify malicious packets and matched malware signatures with identified packets. Ban et al. provide a study on early detection of attacks on darknet. They utilize a time series to characterize the activity level of attack patterns [17]. They also reveal the most prominent attack patterns by employing a clustering algorithm that clusters the attack patterns into groups with the same activities. To provide visual insights into the relationships between clusters, a dimension reduction is employed. Their experimental results indicate the effectiveness and efficiency of the proposed approach in early detection of new attack patterns.

To gain a better understating of the darknet traffic and its parameters in attack identification, Gadhia et al. focused on a comparative analysis over two darknet sensors [18]. Studying total incoming packet, number of source host, targeting destination port for TCP and UDP protocols, they discovered that the darknet sensors have wide difference in incoming traffic characteristics. Fachkha et al. proposed an approach to infer and characterize DNS Distributed Reflection Denial of Service (DRDoS) attacks in dark network [19]. Their work relied on intensity, rate, and geo-location in addition to various network-layer and flow-based features. They employed k-means clustering to identify campaigns of DRDoS Attacks. In another attempt, Fachkha and Debbabi presented a comprehensive survey on darknet and discuss on other trap-based monitoring systems and compare them to darknet [20]. They report case studies on Conficker worm, Sality SIP scan botnet, and the largest amplification attack in 2014 to provide analysis on darknet information. Their work further identifies Honeyd as probably the most practical tool to implement darknet sensors.

Ling et al. proposed and implemented a system to discover and study malicious traffic over Tor [21]. The system uses an intrusion detection system to classify the malicious traffic. Their experimental result reveal approximately 10% of Tor traffic that can trigger alerts of the intrusion detection system. The identified malicious traffic includes P2P traffic, malware traffic, denial-of-service attack traffic, spam, etc. Wang et al. proposed a new person attribute extraction method with the aim of obtaining a comprehensive characterization of malicious users and tracing them [22]. Their method is comprised of block filtration, attribute candidate generation, and attribute candidate verification. Using the extracted information, they analyze sensitive personal information such as Top-K name entities, email domain name, etc. of darknet users.

In [23], three different super-resolution algorithms are used for text recognition in the Darknet. They evaluated their proposed algorithm over five state-of-the-art datasets for text spotting in Tor darknet. Their model achieves a 3.41% of improvement when deep CNN and the rectification network are combined. In [24], the authors utilize unsupervised and self-supervised machine learning methods to infer image semantics from unstructured multimedia data to investigate the content of the Darknet's marketplaces. The evaluation demonstrates how the combination of CNN and LDA models can retrieve documents and images from text and image queries on the Darknet.

In [25], network flow features are used for Tor traffic analysis and multi-level cataloging. Their proposed model can detect the anonymous traffic in different levels of *L1*, *L2*, and *L3* for various platforms such as mobile and PC. The work in [26] and [27] investigates the automatic classification of images on Tor darknet websites. The authors propose a semantic attention keypoint filtering (SAKF) to remove non-significant features at the pixel level of images using a bag of visual words (BoVW) framework to improve the classification accuracy.

3. The Proposed Deep Neural Network

A deep neural network is defined as a tuple $N = (L, C, F)$. $L = \{L_i | 1 < i < M\}$ is a set of M layers where the first layer (L_1) is called input, the last layer (L_M) is called output, and the rest are called hidden layers. C in the tuple is defined as $C = L \times L$ which represents a set of connections between all layers, and $F = \{F_i | 2 < i < M\}$ is a set of functions each of which is used for a non-input layer. Each layer L_i consists of P_{L_i} perceptrons where i^{th} perceptron on layer l is denoted by $p_{i,l}$. For each perceptron $p_{i,l}$ in layer L_i, $1 < i < M$, there are two variables $b_{i,l}$ and $a_{i,l}$ that store the values of the perceptron before and after applying an activation function. Activation function or transfer function decides how the value of a perceptron influences its output [28]. The activation function used for hidden layers in this work is ReLU that is the most well-known activation function for deep neural networks. ReLU changes the perceptron's value based on the Equation 1.

$$a_{i,l} = ReLU(b_{i,l}) = \begin{cases} b_{i,l} \; if \; b_{i,l} \geq 0 \\ 0 \; otherwise \end{cases} \tag{1}$$

Since there is no activation function to be applied on input (L_1) layer, the perceptrons on the first layer are associated with only one value that is $a_{i,l}$. Each layer L_i in the network is associated with a vector space, $V_{L_i} = \mathcal{R}^{P_{L_i}}$, to record the $a_{i,l}$ values of its perceptrons, and V_{L_1} is considered as an input. In fully connected neural networks such as the network used in this work, all perceptrons in layer L_i are connected to the perceptrons in layer L_{i-1}. The connection between perceptron $p_{i,l}$, i^{th} perceptron on layer l, and perceptron $p_{j,l+1}$, j^{th} perceptron on layer l+1, is denoted as $w_{l,i,j}$. Now, the value of a perceptron before activation function is defined as Equation 2.

$$b_{l+1,i} = \beta_{l+1,i} + \sum_{1 \leq j \leq P_l} w_{l,i,j}.a_{l,j} \tag{2}$$

where $\beta_{l+1,i}$ is called bias for i^{th} perceptron on layer l+1. Bias values are tuned during the training, and they help shift the activation function. In Figure 1, we visualize such a network that we proposed to classify darknet traffic. Number of perceptrons in the first layer is equal to the number of non-target features. Number of perceptrons in hidden layers is set based on several experiments on the performance of neural network. Output layer contains four perceptrons for four different classes of Tor, non-Tor, VPN, and non-VPN. Other parameters of the network are listed in Table 1.

Table 1. List of parameters and their values in our deep neural network

Parameter name	Parameter value
Activation Function (Hidden layers)	ReLU
Activation Function (Output layer)	Softmax
Loss Function	Sparse Categorical Cross Entropy
Optimizer	Adam
Epochs	100
Batch Size	64
Validation split	33%

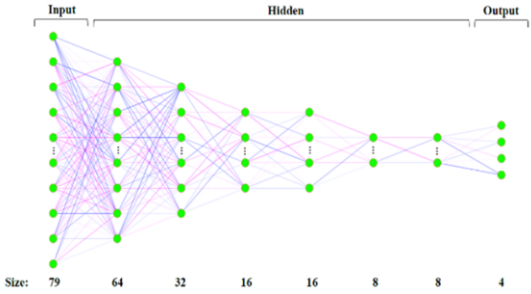

Figure 1. The proposed deep neural network

1. **Activation**: As mentioned earlier, the activation function decides how the weighted sum of input to a perceptron forms its output and eventually the network's output. In this work, we apply ReLU activation function for all hidden layers. To obtain a distribution over the 4 classes in darknet traffic classification, Softmax activation function is used for the output layer. Equation (3) indicates how the Softmax function works for the input vector *a* [29].

$$softmax(a) = \frac{e^{a_i}}{\sum_{1 \le j \le K} e^{a_j}} \tag{3}$$

where K is the number of classes (4 in our problem), e^{a_i} is the standard exponential function for the input vector, and e^{a_j} is the standard exponential function for the output vector.

2. **Loss**: Deep neural networks are trained based on stochastic gradient descent or its variants. Loss is the prediction error of the network while calculating and updating the weights. The loss function, sparse categorical cross entropy in this work, is used to calculate loss value of the prediction. Equation 4 indicates how the sparse categorical cross entropy is defined.

$$L(w) = \frac{1}{N}\sum_{k=1}^{N}[O_k \log(\widehat{O_k}) + (1 - O_k)\log(1 - \widehat{O_k}) \tag{4}$$

In the equation 4, w indicates the weight vector of the neural network, O_k indicates the true labels of the data, and $\widehat{O_k}$ represents the labels predicted by the network. N also indicates the input size.

3. **One hot encoder**: This function is used to transform all the categorical data into numerical form which can help the network to have a better prediction [8].

4. **Dropout**: This function is used to prevent overfitting of the network. A dropout layer randomly sets the value of input perceptrons to zero with a frequency of rate (τ) at each step during training the network. Non-zero inputs are scaled up by $\frac{1}{1-\tau}$ such that the sum of all input perceptrons remains the same. In the experiments, we use $\tau = 0.2$ which produces the best results for this problem.
5. **Optimizer**: In this work, we utilize Adam gradient-based optimization algorithm to update the weights of the network during the training phase [30].
6. **Normalization**: To avoid bias the model towards features with high values, normalization is used to transform the features' values into a decimal in the range of [0, 1]. In this work, we utilize min-max normalization function, *N*, that works based on Equation 5.

$$\forall f \ in \ F: N(f) = \frac{f - f_{min}}{f_{max} - f_{min}} \tag{5}$$

where *F* is the set of all features in the data, *f* indicates a feature, and f_{min} and f_{max} are the minimum and maximum values of *f*.

7. **Adaptive learning rate**: Each time the network weights are updated during training, learning rate hyperparameter is used to control size of moving towards a minimum of the loss function. An initially large learning rate value helps the model accelerate training and gradually reducing the learning rate helps the model learn complex patterns in the data. In this study, we use exponential decay function [31] that is shown in Equation 6.

$$lr = lr_{in} \times kt \tag{6}$$

In the equation above, lr_{in} is the initial learning rate value, k is a hyperparameter to control the reduction amount, and t is the iteration number. To set the initial learning rate, we consider changes of learning rate in response to the loss value during the training phase.

4. Experiments

We first discuss on the dataset used for darknet traffic classification. This data is called DIDarknet and is a recently collected benchmark dataset of 141,529 traffic instances with 79 features[2] where 55 features have integer values, and the rest are decimal. The dataset contains four classes of Tor (1,392 samples), VPN (22,919 samples), non-Tor (93,355 samples), and non-VPN (23,863 samples) where class 1 (Non-Tor) and class 4 (VPN) are the majority and the minority, respectively.

To evaluate our model, we randomly allocate 20% of the data for test and 80% for training and validation. The training phase is used to train the model and initialize the weights of the neural network. As Table 1 indicates, 33% of the train data is allocated for validation. The validation set is used after training the model to tune the hyperparameters with the aim of improving the model's accuracy. Figure 2 shows values of loss and accuracy for both train and validation sets.

[2] For the description of all features, please refer to the original website of the dataset: https://www.unb.ca/cic/datasets/darknet2020.html

The loss metric is calculated for both training and validation and shows the sum of errors made on each instance in the training and validation sets. In other words, loss measures how the distribution of a class labels are different from the distribution of labels predicted for the class's instances. During the training phase, loss is used to tune the weights of the neural network and each iteration of the optimization, the loss values imply how well the model behaves. In this work, we employ sparse categorical cross entropy as the loss function.

The accuracy metric is used to measure the model's performance after setting the parameters. It is the measure of how accurate the model predicts the true data. Plot of loss values in Figure 2 demonstrates that after training the model, loss decreases on the validation set which implies the neural network performs better after tuning the model's parameters. According to the plot of accuracy, the neural network learns to predict with an accuracy over 95%, and the prediction accuracy on the validation set is almost the same.

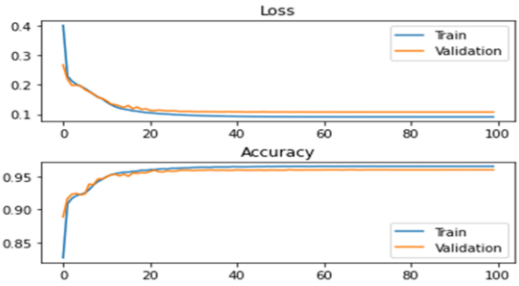

Figure 2. Loss and accuracy for training and validation

As we discussed before, we control the size of the movements on the search space by an adaptive learning rate. To empower our model to use an effective learning rate, we choose its initial value by examining the effects of various learning rates on the loss function. Figure 3 shows the plot of changes where for learning rate values greater than 10^{-3} and close to 10^{-2}, loss is minimum. By guess and check iterations over the values in this range, we set the initial learning rate to 3×10^{-3}.

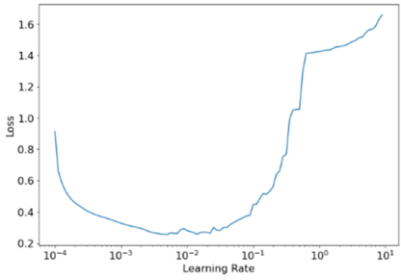

Figure 3. Loss Vs. Learning rate

We now evaluate the deep neural network's performance using the test data. The data is fed into the input layer, and the evaluation metrics listed in Table 2 are calculated. Since the data in this work contains more than two classes, all the evaluation metrics in Table 2 are used as multi-class metrics. Also, for an easier interpretation, values for the first four metrics are reported as percentage.

Table 2. Evaluation metrics used to evaluate the deep neural network's performance

Metric	Equation	Metric variables	Value
Accuracy	$\frac{TP+TN}{P+N}$	P: size of class P N: size of class N TP: No. of samples correctly classifies as P TN: No. of samples correctly classifies as N FP: No. of samples incorrectly classifies as P FN: No. of samples incorrectly classifies as N	96.07
Precision	$\frac{TP}{TP+FP}$		96.08
Recall	$\frac{TP}{TP+FN}$		96.12
F1	$\frac{2TP}{2TP+FP+FN}$		96.06
Kappa	$\frac{P_{obs}-P_{exp}}{1-P_{exp}}$	P_{obs}: empirical probability of agreement on the label assigned to any sample P_{exp}: expected agreement when labels are assigned randomly	0.9225

Accuracy is one of the well-known metrics used to evaluation classification performance of machine learning techniques. Accuracy reports total number of correct predictions to the total number of all samples. In case of imbalanced data, accuracy is not a proper metric since the ratio can be biased towards the majority class. Precision is another classification metric that indicates how precise the classifier is in predicting true samples of each class. In other words, precision reports what portion of all samples classified in a class truly belong to that class. In contrast, recall is a metric to represent the percentage of samples in a class that are correctly predicted by the classifier. F1 is the harmonic mean of precision and recall and indicates the quality of classification.

In contrast to the stet-of-the-art model, DIDarknet, [8] with accuracy of 85%, our model notably outperforms based on four well-known evaluation metrics. To gain a better understanding of the model performance, the confusion matrix of the model is shown in Figure 4. The diagonal values indicate how well the model predicts true data in each class while non-diagonal values on each row indicate number of instances in a class that are incorrectly classified in other classes.

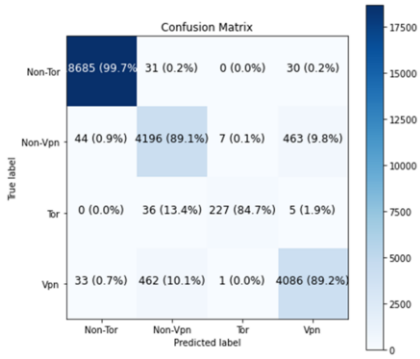

Figure 4. Confusion matrix of the classification

Kappa is a statistical score in [0,1] that reports how much two annotators (true labels and predicted labels) agree on the labels assigned to the samples in a classification problem. The following guideline published in [32] can be used to interpret the Kappa metric: value 0.00 to 0.20 is considered slight agreement; 0.21 to 0.40 is fair agreement; 0.41 to 0.60 is moderate agreement; 0.61 to 0.80 is substantial agreement; and 0.81 to 1.00 is almost perfect agreement. According to the value reported for Kappa in Table 2,

the deep neural network presents a perfect agreement between the true and predicted labels in darknet traffic classification.

5. Conclusion and Future Work

Darknet traffic classification plays an important role in detection of cyberattacks and malicious activities on the Internet. This work proposes a deep neural network for darknet traffic classification. We utilize a recently published benchmarked dataset of Web traffic that contains both anonymized VPN and Tor activity instances to represent the real darknet traffic. The main purpose is to classify the darknet traffic into four classes of non-Tor, non-VPN, Tor, and VPN. The state-of-the-art deep neural models proposed for the problem employ feature selection/extraction prior to classification to reduce number of features. However, our model can identify salient features in traffic data during training the network. It also handles the high-class imbalance in the data without any balancing technique prior to classification. Based on different types of evaluation metrics, our model outperforms the related work with a notable difference of 10% in classification accuracy and Kappa value of 0.92.

To extend this work, we plan to evaluate the performance of deep neural network in classification of dark hidden services regarding their textual information and their structural identity [33]. Also, creating and publishing another dataset that contains more balanced data of Tor and VPN traffic can be another direction for future work. Regarding rapidly changing traffic of darknet, we can also expand this work further by studying the evolution of the traffic and its features over time and implement a deep neural network to classify big data of darknet traffic over time.

References

[1] D. Goldschlag, M. Reed and P. Syverson, "Onion routing for anonymous and private internet connections," Communications of the ACM, vol. 42, no. 2, p. 5, 1999.
[2] M. W. Al-Nabki, F. Eduardo and A. Enrique, "Torank: Identifying the most influential suspicious domains in the tor network," Expert Systems with Applications, vol. 123, pp. 212-226, 2019.
[3] M. Zabihimayvan, R. Sadeghi, D. Doran and M. Allahyari, "A broad evaluation of the Tor English content ecosystem," in Proceedings of the 10th ACM Conference on Web Science, 2019.
[4] N. Hashimoto, S. Ozawa, T. Ban, J. Nakazato and J. Shimamura, "A Darknet Traffic Analysis for IoT Malwares Using Association Rule," in Conference on Big Data and Deep Learning, Procedia Computer Science, 2018.
[5] K. Kanemura, K. Toyoda and T. Ohtsuki, "Identification of darknet markets' bitcoin addresses by voting per-address classification results," in 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 2019.
[6] Y. Hu, F. Zou, L. Li and P. Yi, "Traffic Classification of User Behaviors in Tor, I2P, ZeroNet, Freenet," in 19th International Conference on Trust, Security and Privacy in Computing and Communications, 2020.
[7] P. Choorod and G. Weir, "Tor Traffic Classification Based on Encrypted Payload Characteristics," in National Computing Colleges Conference, 2021.
[8] A. Habibi Lashkari, G. Kaur and A. Rahali, "DIDarknet: A Contemporary Approach to Detect and Characterize the Darknet Traffic using Deep Image Learning," in The 10th International Conference on Communication and Network Security, 2020.
[9] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," Journal of Big Data, vol. 6, no. 1, pp. 1-54, 2019.
[10] V. Ganganwar, "An overview of classification algorithms for imbalanced datasets," International Journal of Emerging Technology and Advanced Engineering, vol. 2, no. 4, pp. 42-47, 2012.
[11] J. P. Early, C. E. Brodley and C. Rosenberg, "Behavioral authentication of server flows," in Annual Computer Security Applications Conference, 2003.
[12] W. H. Turkett Jr, A. V. Karode and E. W. Fulp, "In-the-Dark Network Traffic Classification Using Support Vector Machines," in Association for the Advancement of Artificial Intelligence, 2008.

[13] L. Bernaille, R. Teixeira and K. Salamatian, "Early application identification," in Proceedings of the ACM CoNEXT conference, 2006.
[14] . L. Bernaille and R. Teixeira, "Early recognition of encrypted applications," in International Conference on Passive and Active Network Measurement, 2007.
[15] C. V. Wright, F. Monrose and G. M. Masson, "On inferring application protocol behaviors in encrypted network traffic," Journal of Machine Learning Research, vol. 7, no. 12, 2006.
[16] H. Nishikaze, S. Ozawa, J. Kitazono, T. Ban, J. Nakazato and J. Shimamura, "Large-scale monitoring for cyber attacks by using cluster information on darknet traffic features," Procedia Computer Science, vol. 53, pp. 175-182, 2015.
[17] T. Ban, S. Pang, M. Eto, D. Inoue, K. Nakao and R. Huang, "Towards early detection of novel attack patterns through the lens of a large-scale darknet," in Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress, 2016.
[18] F. Gadhia, J. Choi and B. Cho, "Comparative analysis of darknet traffic characteristics between darknet sensors," in International Conference on Advanced Communication Technology, 2015.
[19] C. Fachkha, E. Bou-Harb and M. Debbabi, "Inferring distributed reflection denial of service attacks from darknet," Computer Communications, vol. 62, pp. 59-71, 2015.
[20] C. Fachkha and M. Debbabi, "Darknet as a source of cyber intelligence: Survey, taxonomy, and characterization," IEEE Communications Surveys and Tutorials, vol. 18, no. 2, pp. 1197-1227, 2015.
[21] Z. Ling, J. Luo, K. Wu, W. Yu and X. Fu, "TorWard: Discovery, Blocking, and Traceback of Malicious Traffic Over Tor," IEEE Transactions on Information Forensics and Security, vol. 10, no. 12, pp. 2515-2530, 2015.
[22] M. Wang, X. Wang, J. Shi, Q. Tan, Y. Gao, M. Chen and X. Jiang, "Who are in the Darknet? Measurement and Analysis of Darknet Person Attributes," in Conference on Data Science in Cyberspace (DSC), 2018.
[23] P. Blanco-Medina, E. Fidalgo, E. Alegre and F. Jánez-Martino, "Improving Text Recognition in Tor darknet with Rectification and Super-Resolution techniques," in IET Conference Proceedings. The Institution of Engineering & Technology, 2019.
[24] A. Berman and C. L. Paul, "Making sense of darknet markets: Automatic inference of semantic classifications from unconventional multimedia datasets," in International Conference on Human-Computer Interaction, 2019.
[25] L. Wang, H. Mei and V. S. Sheng, "Multilevel identification and classification analysis of Tor on mobile and PC platforms," IEEE Transactions on Industrial Informatics, vol. 17, no. 2, pp. 1079-1088, 2020.
[26] E. F. Fernandez, R. A. V. Carofilis, F. J. Martino and P. B. Medina, "Classifying Suspicious Content in Tor Darknet," arXiv preprint arXiv:2005.10086, 2020.
[27] E. Fidalgo, E. Alegre, L. Fernández-Robles and V. González-Castro, "Classifying suspicious content in tor darknet through Semantic Attention Keypoint Filtering," Digital Investigation, vol. 30, pp. 12-22, 2019.
[28] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT Press, 2016.
[29] R. Szeliski, Computer vision: algorithms and applications, Springer Science and Business Media, 2010.
[30] F. Chollet, Deep Learning with Python, Simon and Schuster, 2017.
[31] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in Neural networks: Tricks of the trade, Springer, 2012, pp. 437-478.
[32] R. Landis and K. Gary, "The measurement of observer agreement for categorical data," Biometrics, pp. 159-174, 1977.
[33] M. Zabihimayvan, R. Sadeghi, D. Kadariya and D. Doran, "Interaction of Structure and Information on Tor," in International Conference on Complex Networks and Their Applications, 2020.