9-2022

Dark Web Analytics : A Comparative Study of Feature Selection and Prediction Algorithms

Andrew Alhussan

Izzat Alsmadi

Abdullah Wahbeh

Mohammad A. Al-Ramahi

Ahmad Al-Omari

Dark Web Analytics : A Comparative Study of Feature Selection and Prediction Algorithms

1st Andrew Allhusen
Engineering & Computer Science
Syracuse University
Syracuse, New York, USA
awallhus@syr.edu

2nd Izzat Alsmadi
Dept. of computing and cyber security
Texas A&M, San Antonio
San Antonio, Texas
ialsmadi@tamusa.edu

3rd Abdullah Wahbeh
Dept. of computer science
Slippery Rock University
Slippery Rock, Pennsylvania
Abdullah.wahbeh@sru.edu

4th Mohammad Al-Ramahi
Dept. of computing and cyber security
Texas A&M, San Antonio
San Antonio, Texas
mrahman1@tamusa.edu

5th Ahmad Al-Omari
CIS and Security Program
Our Lady of the Lake University
San Antonio, Texas
aal-omari@ollusa.edu

***Abstract*—The value and size of information exchanged through dark-web pages are remarkable. Recently Many researches showed values and interests in using machine-learning methods to extract security-related useful knowledge from those dark-web pages. In this scope, our goals in this research focus on evaluating best prediction models while analyzing traffic level data coming from the dark web.**

Results and analysis showed that feature selection played an important role when trying to identify the best models. Sometimes the right combination of features would increase the model's accuracy. For some feature set and classifier combinations, the Src Port and Dst Port both proved to be important features. When available, they were always selected over most other features. When absent, it resulted in many other features being selected to compensate for the information they provided. The Protocol feature was never selected as a feature, regardless of whether Src Port and Dst Port were available.

***Index Terms*—Darknet, Network traffic, cyber security, machine learning, prediction algorithms**

I. Introduction

The part of the Web that most users are familiar with represent a small percentage of the Internet [1]. Such part is known as the Surface Web [2]. On the other hand, most of the content on the Internet is hidden within what is referred to as Deep Web. The Deep Web compromise content such as companies intranets, private databases, content with limited access [1]. The term Dark Web refers to contents from the Deep Web that are not normally accessed using a Web browser. Instead, access to Dark Web content requires specialized tools and techniques [3]. The Dark Web segment is cindered the most concerning part of the Deep Web since it is used for legitimate purposes as well as other malicious and criminal activities [4], [5]. In general, gaining access to Dark Web content requires deliberate steps that operates strictly anonymously for the service provider as well as for the user [3]. Access to the Dark Web content requires software tools that rely on volunteer computers to rout content in a way that such content could be traced to the original source [4]. Such need for specialized tools, completely contrast with the Surface Web, which could be accessed using known search engines and Web browsers [3]. According to the literature, the research community is mainly interested in Dark Web traffic because it can help learn the differences between malicious and benign traffic [5]. As a result, it is important to be proactive and strive for the faster identification of threats, regardless of its source, in order to make sure the network is more secure [2]. The faster that threats can be identified, whether from the lightnet (i.e., the version of the internet most people are familiar with), the deepnet, or the darknet, the more secure a network can be. To address such important need, data mining techniques, such as supervised learning techniques, can effectively help classify the malicious and benign Dark Web content [6]. As a result, this paper aims to analyze a Dark Web data set obtained online using different classification methods and features selection techniques. Specifically, we aim to answer these research questions:

- Which classification models would be the best to accurately predict the purpose of the darknet traffic , e.g., audio-streaming, VOIP, email, etc.?
- Which features would be the most useful when modeling the data?

The remainder of the paper is organized as follows: the next section provides an overview of existing literature related Dark Web mining. The research design and methodology section present the data collection, preparation, and analysis. The results section highlights the classification results. The discussion section depicts the key findings from the study. The paper concludes with a summary of contributions and limitations.

II. Literature Review

The threats posed by the Dark Web could help provide valuable clues for protecting against the fallout from data breaches. However, the challenge that many researchers face

is making sure to choose the useful signals from "the vast, dynamic and heterogeneous environment" of the Dark Web [1]. According to the literature, data mining technique can help analyze content from Dark Web in order to protect against malicious attacks. Iliadis and Kaifas [7] have employed machine learning algorithms for classifying Darknet traffic. The authors utilized ROC and features analyses for better visualization results. The research experiments utilized the CIC-Darknet2020, and a number of classifiers were trained to both binary (Benign and Darknet) and multiclass classification (Tor, NonTor, VPN and NonVPN). The authors achieved an average prediction accuracy of 98% using Random Forest algorithm for both classification tasks [7]. Ban et. al. [8] have characterized cyberattacks from large dataset in a distributed Darknet using association rules mining. The authors reported frequent item sets using "probed destination ports and probed darknet sensors". Results showed that association rules mining can help support strategic cyberattack countermeasure in different ways. First, statistical analysis of malware-specific rules can help understand the global trend of cyberattacks on the Internet. In addition, strong rules from association rules mining can shed the light into attacking tools natures, which in turn improve the diagnosis process. Finally, knowledge extracted from frequent attacking patterns can increase prediction accuracy of future cyberattacks [8]. Thorat, Thakur, and Yadav [9] proposed a method for classifying and visualizing illegal activities on Dark Web by selecting relevant laws and regulations about each activities and trained the classifiers. Results showed that related illegal activities are mainly related to drugs, child pornography, and weapons. Three classifiers were trained, Naïve Bayes, SVM, and Random Forest. Results from classification tasks showed that SVM and Random Forest achieved accuracy of 100% followed by the Naïve Bayes classifier with an accuracy of 88.9%. Such classification could help define new approaches for illegal activities categorization as well as a proactive approach for monitoring potential illegal activities [9]. He, He, and Li [6] proposed a new method for classifying illegal activities in Dark Web. The authors trained machine learning classifiers by selecting specific laws and regulations related to different types of illegal activities instead of training the classifier using large Dark Web training set. Training was done using a set of documents from the United States on illegal activities on Dark Web. Results showed that TF-IDF feature selection and Naïve Bayes classifier achieved an accuracy of 93.5% in the experimental environment. Such prediction accuracy could help identify illegal activities from Dark Web content which in turn help in detecting and monitoring potential threats in a timely manner [6]. According to the literature and up to the knowledge of research, the literature lacks a comprehensive study to evaluate different features selection approaches and classification algorithms for the prediction and analysis of Dark Web content. This study aims at addressing such limitation by determining the best features that are most helpful when modeling Dark Web traffic as well as determining the best classification models to accurately predict the purpose of Dark Web traffic.

III. RESEARCH METHODOLOGY

The research methodology followed is demonstrated in Fig. 1. Detailed discussion about the methodology is presented in the subsequent sections.

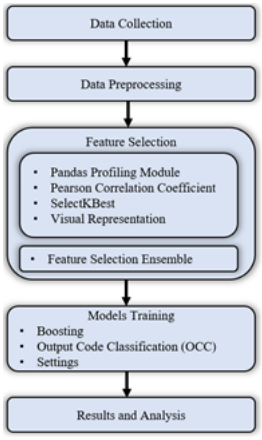

Fig. 1. Our Research Methodology

A. Data Collection

Prior to choosing the dataset, several other datasets were considered. Most datasets covered general internet traffic, but the chosen dataset covers darknet traffic specifically, making it a more specific topic which always draws interest due to its mysterious nature. The chosen dataset is published on kaggle.com [10]. However, it was originally published on the website of the Canadian Institute for Cyber Security. It was created by combing two separate datasets from the Canadian Institute for Cyber Security and the University of New Brunswick. The dataset consists of 141,529 entries each with 85 columns. Each row contains information about a connection made across the darknet, including the IP addresses, packet size, transmission rates, etc.

B. Experiments

Preprocessing activities: Prior to beginning the analysis, the dataset was examined to check for any issues that could obstruct the analysis. Several issues had to be resolved while checking the file. For example, there were two columns with the same name (Label), and one of them was the target column. The non-target column was renamed to Conn Type since it was a more accurate description of its information. The target column was renamed to Purpose. Besides renaming some columns, repeated columns like (Flow ID) was removed. In addition, there were 49 rows that contained NaN or Infinity values for certain columns were also removed. Furthermore, the values of some columns like Src IP, Dst IP, Timestamp, Conn Type, and Purpose were converetd/translated into appropriate values. The dataset after all cleaning actions consists of 141,042 entries each with 85 columns. An analysis was then performed to acquire some basic facts about the dataset. The analysis revealed that there were also 15 columns that

TABLE I
TOP 5 FEATURES ACCORDING TO SELECTKBEST APPROACH

Feature	Is Top?	F Value	P Value
Bwd IAT Total	TRUE	415.8626	$< .001$
Idle Mean	TRUE	2222.0877	$< .001$
Idle Std	TRUE	255.9679	$< .001$
Idle Max	TRUE	2275.9854	$< .001$
Idle Min	TRUE	1457.6256	$< .001$

had constant values and 24,456 duplicate lines. All constant columns were removed. Only one copy of each duplicate row was kept.

Feature selection: With the preprocessing steps completed, the next step was to analyze the available features to determine which would be the most important features. To determine the important features, many different forms of analysis were used.

- The pandas profiling module for Python creates a nicely format report which displays detailed statistics for all features in the dataset and creates charts for visualization. This report provided a good reference for the dataset overall and provided a good indication of which features should be analyzed as a potential important feature.
- Calculating the Pearson Correlation Coefficient was another approach used for evaluating the importance of the features. This was done to check if there were any trends between any two features of the datasets. After exporting the results to an Excel worksheet, the data was marked-up to easily show the interesting data points. Apart from the same feature having a coefficient of 1 with itself, there were four other pairings of features that had a coefficient of 1, which means they are perfectly positively linearly-related. There were no features that had a coefficient of -1, which means they were perfectly negatively linearly-related. However, there were several pairs that were close, and the closest was -0.8602 between Protocol and Fwd Seg Size Min.
- Best features: The next approach was to use (SelectKBest) to determine which features are considered the top K features. In the analysis, the value for K was set to 3, 5, and 10. In addition, the F-Value and the P-Value were also calculated for each feature. The results are shown in the table I, but it only includes the top 5 features

The final approach for the feature selection step was to 4) create a visual representation for the dataset's features. This had two parts. The first part involved creating a scatter plot and a histogram for each feature. This provided a way to see how the information for each feature was distributed.

Feature selection ensemble: The above feature selection results were based on analyzing the dataset without involving classifiers. The next step of feature selection was to analyze the features used by certain classifiers. These features would then be used in different training trials to measure the effect of selecting features. There were four different options explored when evaluating the dataset: evaluate with all features, evaluate with features selected from all features, evaluate with all features except Src Port and Dst Port, and evaluate with selected features from all features except Src Port and Dst Port. There were three classifiers used for the feature selection step: RandomForestClassifier, ExtraTreesClassifier, and DecisionTreeClassifier; each classifier used a max_depth of 4. For each iteration of feature selection, the top 5 features were selected. The table II shows the different features selected based on the available features.

TABLE II
THE SELECTED FEATURES

All Features	All features except Scr Port and Dst Port
Flow Duration	Idle Max
Idle Max	Protocol
Src Port	Bwd Packets/s
Dst Port	Conn Type
Conn Type	Src IP
Idle Mean	Idle Mean
Subflow Fwd Packets	Subflow Fwd Packets
Timestamp	Bwd Packet Length Min
Packet Length Min	Dst IP
	Timestamp
	Average Packet Size

Between the two cases, there are some common features. However, the interesting thing to note is that the number of selected features changes when the Src Port and Dst Port are removed. When the top 5 features were selected from all feature, there was overlap form the classifiers because they agreed on the useful features. When removing these two features and selecting again, although there were some common features still, the classifiers differed more in the features they used.

Training the Models: The training of classifiers (sometimes referred to as "models") was then separated into trials. Each trial consisted of three phases, which related to the type of models being used. The three phases are Boosting, Output Code Classification (OCC), and Settings. The Boosting phase is where different forms of boosting classifiers are compared to the standard classifiers, which are used as the base estimators for the boosting classifiers. The OCC phase uses the OutputCodeClassifier with various different base classifiers. The Settings phase used compares different versions of the standard classifiers by manipulating the classifier's settings. The trial was completed for 25/75 train/test splits. All classifiers are graded on the same metrics: accuracy, precision, and recall.

IV. RESULTS AND DISCUSSION

A. Ensemble classifiers (Boosting Classifiers)

The boosting phase consisted of 1 standard RandomForest classifier, 1 GradientBoosting classifier, 1 XGB classifier, several AdaBoost classifiers (base estimators included RandomForest, ExtraTrees, DecisionTrees, and GradientBoost), and an EnsembleVote classifier, which included all the previous classifiers. The RandomForest classifier was also different from other versions used, since there was no limit on the max depth of the tree. Most other

versions of this classifier limit the max depth to 4. In these trials, the train/test split was at 25/75. Within each trial, the accuracy, precision, and recall have been color-coded, where green indicates the highest scores, followed by blue, yellow, and dark red.

Model	Accuracy	Precision	Recall
All Features - 68 Features			
RandomForest-4	0.980	0.973	0.973
ExtraTrees-4	0.907	0.834	0.879
DecisionTree-4	0.860	0.754	0.858
Kneighbors	0.982	0.979	0.982
Ridge-300	0.999	0.999	0.999
LogisticReg-300-ncg	0.949	0.887	0.952
All Features - No Ports - 66 Features			
RandomForest-4	0.977	0.968	0.970
ExtraTrees-4	0.901	0.829	0.897
DecisionTree-4	0.833	0.736	0.842
Kneighbors	0.985	0.986	0.985
Ridge-300	0.999	0.998	0.998
LogisticReg-300-ncg	0.946	0.890	0.952
Selected Features - 10 Features			
RandomForest-4	0.997	0.996	0.996
ExtraTrees-4	0.924	0.860	0.922
DecisionTree-4	0.882	0.805	0.840
Kneighbors	0.972	0.967	0.978
Ridge-300	0.999	0.999	0.999
LogisticReg-300-ncg	0.950	0.896	0.953
Selected Features - No Ports - 11 Features			
RandomForest-4	0.995	0.993	0.994
ExtraTrees-4	0.915	0.861	0.908
DecisionTree-4	0.852	0.744	0.843
Kneighbors	0.992	0.993	0.991
Ridge-300	0.997	0.997	0.997
LogisticReg-300-ncg	0.955	0.903	0.956

Fig. 2. ENSEMBLE CLASSIFIERS RESULTS

Figure 2 shows the results from ensembles classifiers. The performance of the models varied based on the features being used. The AdaBoost(ExtraTrees) classifier had the worst performance in three of the four trials, and in the one other case, it was the second worst performer. AdaBoost (GradientBoosting) was a strong performer in most trials. However, it struggled in the case where the selected features did not include the Src Port and Dst Port. RandomForest had a steady performance. It never took first place among the classifiers, but it was always near the top, similar to AdaBoost (DecisionTree). The EnsembleVote was a contender for top-performing classifier, but it fell just short every trial. Across all trials the metric with the highest standard deviation was precision (0.1023), and accuracy (0.0793) and recall (0.0801) were close to each other. Despite the large different in the number of features used in the trials, the accuracy, precision, and recall did not show great variations, though there are some lower percentages. The EnsembleVote classifier met expectations, since it was expected to score high given it had the collective knowledge of several other classifiers. The RandomForest classifier, although accurate, did perform slightly below the anticipated benchmarks.

B. Output Code Classifiers

The OCC trials use several classifiers with similar settings to those in the boosting trials. These classifiers included 1 standard classifier for each of the following types: RandomForest, ExtraTrees, DecisionTree, KNeighbors, Ridge, and LogisticRegression. The tree-based classifiers all used a max depth of 4. The Ridge and LogisticRegression classifiers had a max_iter parameter that was set to 300. In these trials, the train/test split was at 25/75. Within each trial, the accuracy, precision, and recall have been color-coded, where green indicates the highest scores, followed by blue, yellow, and dark red.

Model	Accuracy	Precision	Recall
All Features - 68 Features			
RandomForest-4	0.757	0.561	0.650
ExtraTrees-4	0.653	0.434	0.498
DecisionTree-4	0.843	0.696	0.687
Kneighbors	0.907	0.886	0.888
Ridge-300	0.675	0.494	0.649
LogisticReg-300-ncg	0.288	0.137	0.091
All Features - No Ports - 66 Features			
RandomForest-4	0.725	0.536	0.508
ExtraTrees-4	0.606	0.418	0.586
DecisionTree-4	0.843	0.702	0.690
Kneighbors	0.910	0.891	0.895
Ridge-300	0.657	0.484	0.619
LogisticReg-300-ncg	0.302	0.144	0.088
Selected Features - 9 Features			
RandomForest-4	0.827	0.646	0.712
ExtraTrees-4	0.699	0.503	0.480
DecisionTree-4	0.843	0.697	0.687
Kneighbors	0.985	0.981	0.982
Ridge-300	0.609	0.427	0.414
LogisticReg-300-ncg	0.386	0.201	0.137
Selected Features - No Ports - 11 Features			
RandomForest-4	0.778	0.596	0.558
ExtraTrees-4	0.618	0.471	0.543
DecisionTree-4	0.843	0.701	0.689
Kneighbors	0.988	0.985	0.985
Ridge-300	0.563	0.397	0.385
LogisticReg-300-ncg	0.327	0.189	0.080

Fig. 3. OUTPUT CODE CLASSIFIERS RESULTS

As shown in Figure 3, the performance of the OCC with the various underlying classifiers produced much more varied results than the boosting trials, but the best and worst performers stayed consistent. The KNeighbors had the highest accuracy in every trial. For most of the trials, the accuracy was nearly the same. However, the last trial using the selected features that excluded the port had the highest accuracy (0.988). Finally, the LogisticRegression had the lowest accuracy in every trial. Across all trials the metric with the standard deviation for accuracy (0.2053), precision (0.2337), and recall (0.2450) were somewhat close to each other. These results were much more varied with values ranging from the high-20% to the very-high-90%. The training times for all model versions were quick, talking only a few seconds, with the exception of the OCC (LogisticRegression), which took just over 3 minutes when using a larger set of features. The boosting-based models provided better performance than the OCC-based models. The boosting-based models had an overall higher accuracy (96% vs. 68%) than the OCC-based models, which were also outscored in average precision and recall.

C. Experiments with Individual classifiers

The final trial phase was composed of individual classifiers. There were several classifiers that had appear in earlier trials,

such as RandomForest, ExtraTrees, DecisionTree, KNeighbors, and Ridge. There were also two additional classifiers, GaussianNB and the Multi-layer Perceptron (MLP) neural network. This phase is labeled as the "Settings" phase because it focuses on how the settings for the individual classifiers can make an impact on their performance metrics. ExtraTrees, DecisionTree, and GaussianNB only had the base versions. ExtraTrees and DecisionTree had a max_depth of 4. All other classifiers had multiple versions. RandomForest had two versions, one having a max_depth of 4 and the other had a max_depth of 8. KNeighbors had three versions. The base versions used 5 neighbors. The other versions used twice and half that number, i.e., 10 and 2 respectively. MLP had four versions: one pair had max_iter set to 300 and the other pair had max_iter set to 600. In each pair, one had early_stopping enabled. The early_stopping tells the model to stop training when the internal cross-validation scores no longer improve. In these trials, the train/test split was at 25/75. Within each trial, the accuracy, precision, and recall have been color-coded, where green indicates the highest scores, followed by blue, yellow, and dark red Figure 4 shows the results from all trials for the individual classifiers. The MLP performance was unexpectedly poor. The accuracy was consistently poor across all versions of the classifier. However, the early_stopping settings proved useful in increasing the accuracy. GaussianNB also performed poorly across all trials. The remaining classifiers, such as RandomForest, DecisionTree, KNeighbors, had fairly consistent performances across all trials. However, there was significant improvement seen in KNeighbors when selected features were used in favor of all available features. The highest accuracy scores were achieved by KNeighbors across all trials.

Model	Accuracy	Precision	Recall
All Features - 68 Features			
RandomForest-4	0.756	0.558	0.526
RandomForest-8	0.930	0.900	0.905
ExtraTrees-4	0.652	0.433	0.628
DecisionTree-4	0.845	0.699	0.689
KNeighbors-5	0.912	0.889	0.893
KNeighbors-10	0.898	0.869	0.880
KNeighbors-2	0.912	0.881	0.912
RidgeClassifier-300	0.674	0.492	0.634
GaussianNB	0.260	0.175	0.113
MLP-300	0.387	0.243	0.322
MLP-300-early	0.375	0.212	0.308
MLP-600	0.439	0.309	0.471
MLP-600-early	0.374	0.226	0.230
All Features - No Ports - 66 Features			
RandomForest-4	0.730	0.541	0.520
RandomForest-8	0.911	0.885	0.899
ExtraTrees-4	0.606	0.419	0.461
DecisionTree-4	0.841	0.696	0.687
KNeighbors-5	0.912	0.893	0.899
KNeighbors-10	0.903	0.879	0.890
KNeighbors-2	0.921	0.898	0.921
RidgeClassifier-300	0.660	0.487	0.633
GaussianNB	0.262	0.177	0.117
MLP-300	0.431	0.292	0.421
MLP-300-early	0.373	0.225	0.220
MLP-600	0.466	0.306	0.398
MLP-600-early	0.303	0.258	0.267
Selected Features - 9 Features			
RandomForest-4	0.826	0.641	0.688
RandomForest-8	0.969	0.957	0.959
ExtraTrees-4	0.694	0.500	0.478
DecisionTree-4	0.846	0.701	0.689
KNeighbors-5	0.985	0.980	0.981
KNeighbors-10	0.979	0.972	0.974
KNeighbors-2	0.988	0.982	0.987
RidgeClassifier-300	0.607	0.425	0.411
GaussianNB	0.276	0.167	0.070
MLP-300	0.236	0.201	0.151
MLP-300-early	0.285	0.129	0.126
MLP-600	0.219	0.139	0.109
MLP-600-early	0.289	0.175	0.142
Selected Features - No Ports - 11 Features			
RandomForest-4	0.779	0.612	0.678
RandomForest-8	0.957	0.949	0.950
ExtraTrees-4	0.624	0.472	0.553
DecisionTree-4	0.847	0.705	0.694
KNeighbors-5	0.988	0.985	0.985
KNeighbors-10	0.983	0.980	0.982
KNeighbors-2	0.988	0.983	0.989
RidgeClassifier-300	0.559	0.394	0.449
GaussianNB	0.281	0.167	0.071
MLP-300	0.175	0.161	0.213
MLP-300-early	0.288	0.133	0.083
MLP-600	0.218	0.131	0.116
MLP-600-early	0.323	0.213	0.165

Fig. 4. INDIVIDUAL CLASSIFIERS RESULTS

V. CONCLUSION

Machine learning techniques combined with proper feature selection approaches can effectively help classify the malicious and benign Dark Web content. In this study, different feature selection approaches were utilized to determine best features, from Dark Web traffic data, that could help improve the prediction accuracy of malicious and benign activities. The study utilized four different sets of features: All Features (68 features), All Features no Ports (66 features), Selected Features (9 features), Selected Features no Ports (11 features). The selected features sets were all used for training and testing three groups of classifiers using 25/75 training/testing splits. The three groups of classifiers were boosting classifiers, output code classifiers, and individual classifiers. Overall, the boosting classifiers performed the best across different features sets followed by the output code classifiers, and individual classifiers. Within the boosting classifiers, the performance was pretty much comparable across all features set. Within the output code classifiers, overall, the classifiers achieved the best performance using the Selected Features (9 features) and Selected Features no Ports (11 features) sets followed by the All Features (68 features) and All Features no Ports (66 features) sets. Finally, within the individual classifiers, the classifiers achieved the best performance using the Selected Features (9 features) set followed by the Selected Features no Ports (11 features) set, All Features (68 features) set, and All Features no Ports (66 features) set respectively. Regardless the feature set, the Ridge-300 classifiers achieved the highest prediction accuracy among the boosting classifiers followed by the Kneighbors and RandomForest-4 classifiers. Among the output code classifiers the Kneighbors classifier achieved the highest predicition accuracy followed by the DecisionTree-4 classifier. Finally, among the individual classifiers, the performance of the KNeighbors-5, KNeighbors-10, KNeighbors-2, and RandomForest-8 classifiers vary from one feature set to another. However, overall, they have achieved the highest

prediction accuracy.

REFERENCES

[1] N. Tavabi, N. Bartley, A. Abeliuk, S. Soni, E. Ferrara, and K. Lerman, "Characterizing Activity on the Deep and Dark Web," in Companion Proceedings of The 2019 World Wide Web Conference, San Francisco USA, May 2019, pp. 206–213. doi: 10.1145/3308560.3316502.

[2] D. Rathod, "Darknet Forensics," Int. J. Emerg. Trends Technol. Comput. Sci. IJETTCS, vol. 6, p. 77, Jul. 2017.

[3] M. Schafer, M. Fuchs, M. Strohmeier, M. Engel, M. Liechti, and V. Lenders, "BlackWidow: Monitoring the Dark Web for Cyber Security Information," in 2019 11th International Conference on Cyber Conflict (CyCon), Tallinn, Estonia, May 2019, pp. 1–21. doi: 10.23919/CYCON.2019.8756845.

[4] K. Finklea, "Specialist in Domestic Security March 10, 2017," Dark Web, p. 19.

[5] D. Singh, A. Shukla, and M. Sajwan, "Deep transfer learning framework for the identification of malicious activities to combat cyberattack," Future Gener. Comput. Syst., vol. 125, pp. 687–697, Dec. 2021, doi: 10.1016/j.future.2021.07.015.

[6] S. He, Y. He, and M. Li, "Classification of Illegal Activities on the Dark Web," in Proceedings of the 2019 2nd International Conference on Information Science and Systems, Tokyo Japan, Mar. 2019, pp. 73–78. doi: 10.1145/3322645.3322691.

[7] L. A. Iliadis and T. Kaifas, "Darknet Traffic Classification using Machine Learning Techniques," in 2021 10th International Conference on Modern Circuits and Systems Technologies (MOCAST), Jul. 2021, pp. 1–4. doi: 10.1109/MOCAST52088.2021.9493386.

[8] T. Ban, M. Eto, S. Guo, D. Inoue, K. Nakao, and R. Huang, "A study on association rule mining of darknet big data," in 2015 International Joint Conference on Neural Networks (IJCNN), Jul. 2015, pp. 1–7. doi: 10.1109/IJCNN.2015.7280818.

[9] H. Thorat, S. Thakur, and A. Yadav, "Categorization of Illegal Activities on Dark Web using Classification," vol. 07, no. 05, p. 6, 2020.

[10] P. Friedrich, "CIC-Darknet2020 Internet Traffic," Sep. 24, 2020. https://www.kaggle.com/peterfriedrich1/cicdarknet2020-internet-traffic (accessed Aug. 20, 2021).