Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.039374
Article

Detecting and Classifying Darknet Traffic Using Deep Network Chains

Amr Munshi[1,2,*], Majid Alotaibi[1,2], Saud Alotaibi[2,3], Wesam Al-Sabban[2,3] and Nasser Allheeib[4]

[1]Computer Engineering Department, Umm Al-Qura University, Mecca, Saudi Arabia
[2]Smart Lab, Umm Al-Qura University, Mecca, Saudi Arabia
[3]Information Systems Department, Umm Al-Qura University, Mecca, Saudi Arabia
[4]College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia
*Corresponding Author: Amr Munshi. Email: aaamunshi@uqu.edu.sa
Received: 25 January 2023; Accepted: 12 April 2023; Published: 26 May 2023

Abstract: The anonymity of the darknet makes it attractive to secure communication lines from censorship. The analysis, monitoring, and categorization of Internet network traffic are essential for detecting darknet traffic that can generate a comprehensive characterization of dangerous users and assist in tracing malicious activities and reducing cybercrime. Furthermore, classifying darknet traffic is essential for real-time applications such as the timely monitoring of malware before attacks occur. This paper presents a two-stage deep network chain for detecting and classifying darknet traffic. In the first stage, anonymized darknet traffic, including VPN and Tor traffic related to hidden services provided by darknets, is detected. In the second stage, traffic related to VPNs and Tor services is classified based on their respective applications. The methodology of this paper was verified on a benchmark dataset containing VPN and Tor traffic. It achieved an accuracy of 96.8% and 94.4% in the detection and classification stages, respectively. Optimization and parameter tuning were performed in both stages to achieve more accurate results, enabling practitioners to combat alleged malicious activities and further detect such activities after outbreaks. In the classification stage, it was observed that the misclassifications were due to the audio and video streaming commonly used in shared real-time protocols. However, in cases where it is desired to distinguish between such activities accurately, the presented deep chain classifier can accommodate additional classifiers. Furthermore, additional classifiers could be added to the chain to categorize specific activities of interest further.

Keywords: Darknet; darknet traffic; deep network chains; Internet traffic

Nomenclature

VPN	Virtual Private Network
Tor	Onion routing
CNN	Convolutional Neural Network
P2P	Peer-to-peer

VoIP	Voice over Internet Protocol
CC	Classifier chains
TP	True positive
TN	True negative
FP	False positive
FN	False-negative
ROC	Receiver operating characteristic
AUC	The Area under the curve
N	Number of training entities
y	Ground-true set of class labels
y^i	True labels for the i-th training entity
$\hat{y}^i$	Detected label for the i-th training entity

1 Introduction

The integration of the Internet into our daily lives is more evident now than ever before. It has revolutionized our day-to-day activities and led to groundbreaking changes in many aspects, such as communication, entertainment, education, transportation, and health care. While the Internet conveniently enables numerous information services, it presents various risks, which include incessant attacks by a wide variety of network-based threats. A common method to identify such threats is monitoring incoming traffic from suspicious network addresses, known as the "darknet." The detection and categorization of Internet network traffic [1] is considered a specialized solution that can effectively deal with critical system infrastructure and monitor various cyber threat activities [2]. Many organizations collect web traffic data to improve their security. The data are then analyzed by being correlated with the presented services. Analyzing of this data can potentially secure those services, guarantee the delivery of critical data, optimize intrusion prevention, and detect malicious activities after outbreaks [3–5]. In [6], a compelling novel approach to protecting users' privacy against sophisticated traffic analysis attacks is proposed. Also, [7] presents a robust dynamic network traffic scheme to defend against malicious attacks. Thus, classifying of darknet traffic is essential for real-time applications such as the timely monitoring of malware before attacks occur.

Anonymous communication attempts to hide services on the Internet with the to secure the communication line from censorship circumvention. Virtual Private Networks (VPNs) and onion routing (Tor) are examples of anonymous communication. VPNs and Tor have similar tunneling communication frameworks. Tor offers greater anonymity than a standard web browser by encrypting web traffic through three encryption layers; however, it is not entirely secure as it is subject to information disclosure, eavesdropping, and man-in-the-middle attacks from malicious exit nodes. Also, both utilize encryption technologies to maintain the integrity of the transferred data. VPNs and Tor are known as representative examples of darknet traffic. More than two million worldwide users directly connect with Tor clients, and a significant number of hidden services are related to suspicious activities [8]. The detection of darknet traffic is, therefore, an area of particular research interest.

Artificial intelligence techniques have been successfully used in numerous problems related to network security, such as detecting malicious encrypted traffic [9–16]. The utilized techniques in those studies usually classify network activities into two main classes, namely, legitimate and abnormal intrusion, to occur whenever there is a deviation from normal activities. Furthermore, many works utilize specific features of the network traffic to guide traffic classification. In [17], an approach to identifying encrypted traffic is presented by employing packet header and statistical flow feature sets, which achieved high accuracy without inspecting the payload, IP addresses, and port numbers. Furthermore, detecting encrypted traffic covering VPN and Tor traffic is of interest. In [18], an

approach to characterizing encrypted traffic into categories, such as browsing, streaming, file transfer, etc., is introduced based on the traffic type. Also, a novel approach is presented in [19] for classifying encrypted VPN Internet traffic and identifying applications by transforming basic flow data into an intuitive picture and then applying image classification techniques. A supervised learning approach using correlation-based feature selection and random forest algorithms to detect Tor traffic effectively is presented in [20]. Recently, approaches for detecting VPN and Tor applications together as the real representatives of darknet traffic have become an area of interest. In [21], a dataset that covers VPN and Tor traffic to create a complete dataset that covers a wide range of captured applications and hidden services provided by darknets is presented. Furthermore, the authors present a two-dimensional convolutional neural network (CNN) that utilizes feature selection techniques to characterize darknet traffic, including VPN and Tor applications with high detection rates. This encourages further investigation into methods for effectively detecting and classifying darknet traffic. To do so, this paper presents a two-stage network chain methodology for detecting and classifying darknet traffic. In the first stage, anonymized darknet traffic, including VPN and Tor traffic related to hidden services provided by darknets, is detected. In the second stage, traffic related to VPNs and Tor services is classified into eight classes:

1) Audio streaming
2) Browsing
3) Chat
4) Email
5) File transfer
6) P2P (peer-to-peer)
7) VOIP (Voice over Internet Protocol)
8) Video streaming

The approach in this paper deploys a chain of deep networks to detect and classify darknet traffic while achieving high accuracy that enables practitioners to combat alleged malicious activities and further detect such activities after outbreaks. The primary contributions of this paper are as follows:

- Identifying the capability of detecting encrypted traffic covering either VPN or Tor traffic separately, covering a wide range of captured darknet activities;
- Demonstrating the effectiveness of the chain of deep networks to detect and classify darknet traffic while achieving higher accuracy measures;
- Presenting a deep chain classifier that can accommodate additional classifiers to categorize specific activities further.

The remainder of this paper is structured as follows. Section 2 presents the multilabel deep network chain classifier. The methodology for detecting and classifying darknet traffic is presented in Section 3. The application and evaluation results of the proposed methodology are presented in Section 4. Finally, the conclusions are discussed in Section 5.

2 Multilabel Deep Network Chain Classifier

Multilabel classification is a supervised learning problem in which one entity can be associated with multiple labels. This is opposed to the traditional problem of single-label classification in which each entity is associated with a single class label. Recently, classifier chain (CC) [22] methods have become a practical approach to multilabel learning problems. The advantage of CCs is that they

combine the computational efficiency of binary relevance methods and can take label dependencies into account for classification. A CC trains a binary or multiclass classifier for each label in a prespecified order on the label set. The features used to induce each classifier are extended by the previous labels in the chain—that is, the labels are treated as additional feature attributes to model the conditional dependence between the label and its predecessors. This approach can be adopted for problems related to the detection and classification of entities, where the first model in the chain "detects," and then the next model "classifies." For that, the CC approach is adopted and used in a dedicated formulation concerning other applications. In this formation, a label from a predecessor model is treated as an additional feature attribute; however, only a subset of entities of interest are passed on the chain to the next model to model the conditional dependence between the label and its predecessors. The supervised models considered in the CC are deep networks in which the first deep network "detects." If the entity is of interest, the entity is passed to the following deep network in the chain, which "classifies" the entity. Fig. 1 presents the chain architecture with two deep network models. It should be noted that such architecture can adopt additional models into the chain, and the architecture can be further formulated based on the required application.

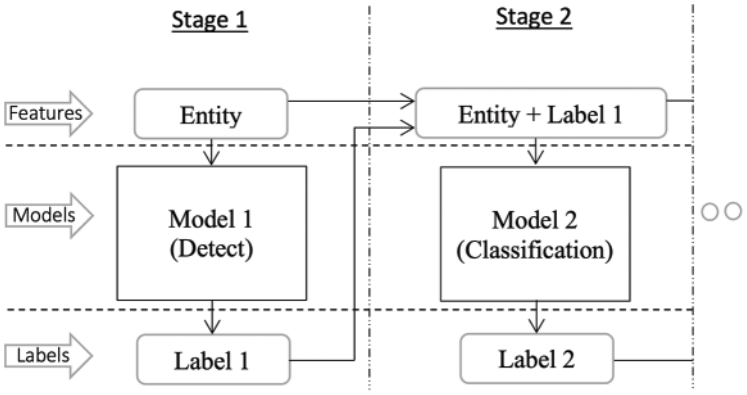

Figure 1: General architecture of the classifier chain

3 Methodology of Detecting and Classifying Darknet Traffic

The main objectives of the methodology presented in this section are: 1) to detect encrypted traffic separately covering both VPN and Tor traffic and a wide range of captured darknet activities, 2) to present an effective chain of deep networks to detect and classify darknet traffic while achieving higher accuracy, and 3) to identify the ability of deep CCs to accommodate additional classifiers in the chain to categorize specific activities of interest.

The detection and classification of darknet traffic are achieved by constructing a chain of two deep network models. The first deep network model is a binary classifier that classifies the entities into normal or suspicious based on the features of the entity. The latter deep network refers to the features of the suspicious entity and categorizes those that are suspicious into different groups based on the activity to which they are related. The methodology of detecting and classifying darknet traffic can be decomposed into three subsequent steps: 1) data preprocessing, 2) deep network CC building, and 3) result evaluation. Fig. 2 presents the general layout of the methodology. To build a classifier that will be able to detect darknet traffic, the data should include both normal and darknet traffic entities. As shown in Fig. 2, the data are to be preprocessed in the first step so that missing and invalid entities are amended or removed. This results in entities carrying the features of the traffic and two

labels. The first label (Label1) is a binary class label, where an entity is classified as either normal or darknet traffic. If the entity is classified as darknet traffic, a second multiclass label (Label2) classifies the type of service. Once the data are preprocessed and include the required features and labels, a Deepnet that can detect darknet traffic entities based on their features is built in the second step. In cases in which an entity is detected as darknet traffic (Label1), in the second stage, the type of darknet service is classified based on the features of the entity and considering Label1.

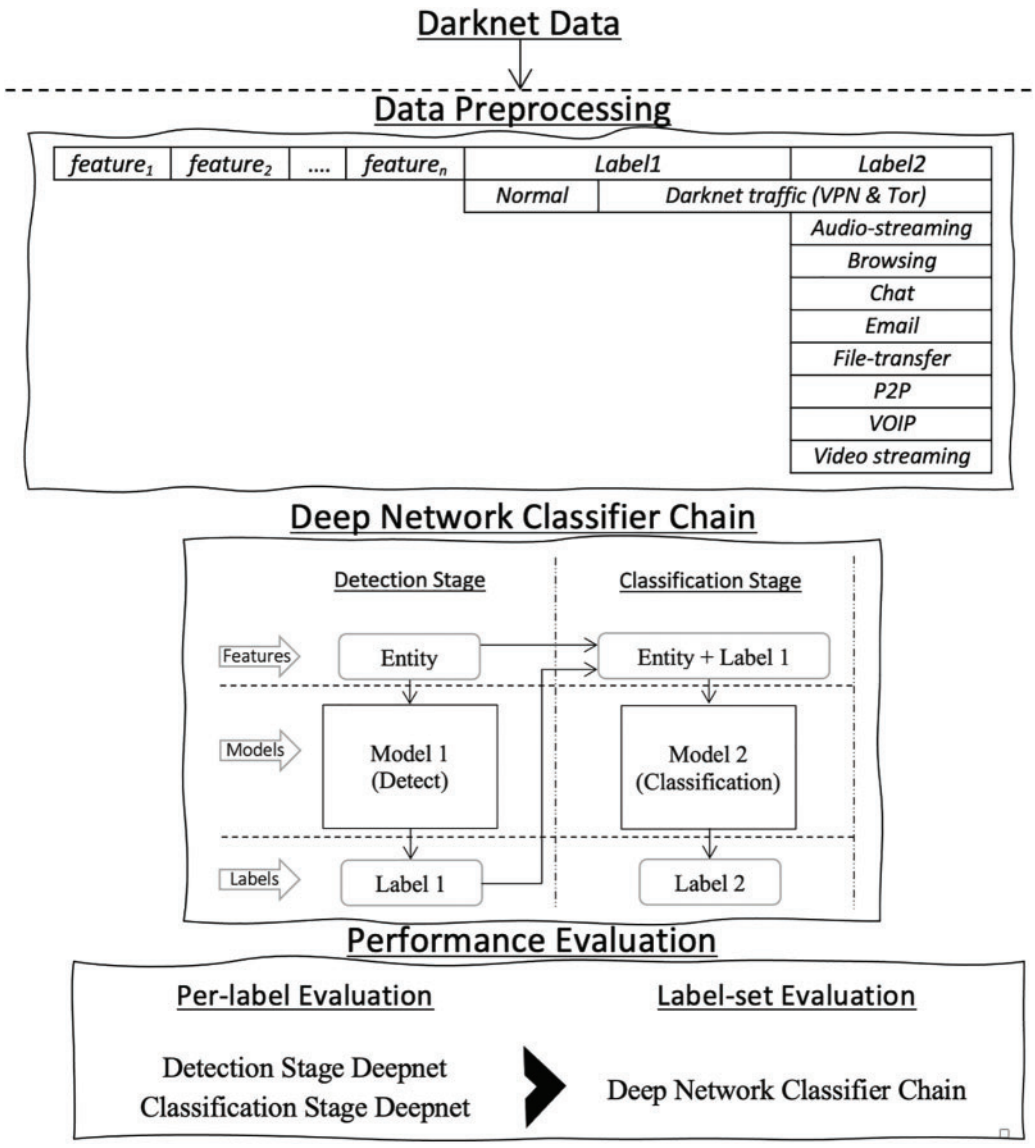

Figure 2: General layout of the methodology

It should be noted that only darknet traffic-detected entities are passed to the classification stage. In the last step, the built models (i.e., the detection and classification stages) are evaluated utilizing evaluation metrics that describe the complete performance of the models. The models can

be continuously improved based on the evaluation results; parameters can be compared to choose the optimal settings. The details of the aforementioned steps are presented in the following subsections.

3.1 Data Preprocessing

A recent dataset that includes both VPN and Tor traffic is used to detect and classify darknet traffic efficiently. The Darknet dataset [21], which combines the ISCXVPN2016 [18] and ISCXTor2017 [23] datasets, is used in this study. The Darknet dataset includes 158,659 entities, of which 24,311 entities are darknet traffic, and 134,348 are normal. Furthermore, darknet traffic entities are categorized into eight hidden services. Each entity's activity is represented by 83 features, including information such as the utilized protocol, total length of the bandwidth packet, bytes/packet flow per second, source IP/port, destination IP/port, timestamp, etc. A darknet traffic entity is an erroneous traffic observed in the empty address space, a collection of globally valid IP addresses that have not been allocated to any hosts or devices. Traffic is not anticipated to enter such a darknet IP space in an ideal, secure network system. The dataset also includes two class labels. The first class label consists of four categorizations: Non-Tor, Non-VPN, Tor, and VPN. The second class label consists of eight traffic activities. It should be noted that network traffic can be collected utilizing data feed platforms such as Apache Spark using Apache Kafka. To formulate the dataset for the construction of a CC that will be able to detect darknet activity and then further classify the type of activity into one of the eight hidden services, each entity in the Darknet dataset is first labeled as "normal traffic" or "darknet traffic" this will be referred to as the *detection stage*. Then, each entity classified as "darknet traffic" is further classified into one of the eight hidden services. This will be referred to as the classification stage. Table 1 presents the number of normal and darknet traffic entities for the *detection stage* and the number of entities for the eight darknet hidden services for the *classification stage* in the Darknet dataset. To detect darknet traffic in general—not specific to known IP addresses—the features are reduced by eliminating flow label features, including flow ID, timestamp, source and destination IP, and ports. This enables us to detect and classify a broad range of darknet activities potentially. The result of this data preprocessing stage is a dataset that includes 158,659 entities, each represented by 64 features, including two-class labels for the detection and classification stages.

Table 1: Number of normal and hidden services activities

Darknet dataset	Detection	Classification
158,659	134,348 (Normal)	n/a
	24,311 (Darknet traffic)	13,284 (Audio streaming)
		263 (Browsing)
		4,541 (Chat)
		582 (Email)
		2,610 (File transfer)
		220 (P2P)
		1,465 (VOIP)
		1,346 (Video streaming)

3.2 Deep Network Classifier Chain Building

In the previous stage, the data were preprocessed for the construction of a CC that will be able to detect darknet activity. The type of activity was further classified into one of the eight hidden services. The deep network CC is constructed by training two deep networks in this stage. The first deep network implements the *detection stage*. Accordingly, the first deep network is a binary classifier that detects darknet traffic. Entities that do not represent darknet activity are classified as normal activities and are not passed on the CC to the next classifier. However, entities detected as darknet traffic are of interest and are passed on the CC to the next classifier to classify the type of activity. Thus, in cases where a new entity contains darknet activity, the classifier should be able to detect the entity as darknet activity. Further, classify each activity based on its features into Audio streaming, Browsing, Chat, Email, File Transfer, P2P, VOIP, or Video streaming.

It should be noted that the class label resulting from the first classifier is added as a feature attribute to the next classifier in the chain. A second multiclass deep network in the CC that implements the *classification stage* is trained to accomplish that task. To build a CC that will achieve the two models, pre-trained deep networks [24] are adopted. The trained deep networks are repurposed by their learned knowledge, which includes the layers, weights, and biases. Furthermore, models that achieve higher classification accuracy are fine-tuned to improve their accuracy and classify entities into the correct class labels. This approach is applied in both the *detection stage* and the *classification stage*. Each Deepnet is evaluated using performance metrics, and the Deepnet that produces the highest accuracy results for each classifier in the CC is chosen. The parameters of the deep network with tuned values for the detection and classification stages are presented in Tables 2 and 3, respectively.

Table 2: Parameters with tuned values for the detection stage

Parameter	Value
Activation function (Hidden layer 1)	RELU
Activation function (Hidden layer 2)	RELU
Layer size (Hidden layer 1)	41
Layer size (Hidden layer 2)	20
Learning rate	0.00073
Optimizer	adam
Batch size	64

Table 3: Parameters with tuned values for the classification stage

Parameter	Value
Activation function (Hidden layer 1)	RELU
Activation function (Hidden layer 2)	RELU
Activation function (Hidden layer 3)	RELU
Layer size (Hidden layer 1)	64
Layer size (Hidden layer 2)	128
Layer size (Hidden layer 2)	64
Learning rate	0.001
Optimizer	adam
Batch size	128

3.3 Performance Evaluation

To evaluate the deep network CC, each classifier is fine-tuned and assessed independently. The accuracy, precision, recall, and F-score metrics are used to assess the performance of the built classifiers. In addition, a performance indicator based on the entities' ground truth and detection probability, namely, the receiver operating characteristic (ROC) curve, is utilized. The ROC curve quantifies the classifier's performance by calculating the area under the curve (AUC).

As mentioned previously, eight class labels are in the second multilabel classifier in the chain (i.e., the *classification stage*). The performance evaluation is computed for every class label, and the final result is based on the overall average of all classes.

In CC problems, it is essential to include multiple and contrasting evaluation measures due to the additional degrees of freedom that the multilabel setting introduces [25]. For that, per-label and label set-based evaluation, which evaluates the label sets of the CC, are utilized in the experimental evaluation.

When a predicted set of class labels ($\hat{y}$) match the ground-true set of class labels (y) exactly (i.e., label set-based evaluation), this is considered the exact match measure is known as the "0/1 loss" measure. In this measure, any label set not detected and classified ideally is given a zero score. N is the number of training entities, y^i is the true label for the i-th training entity, and $\hat{y}^i$ is the detected label for the i-th training entity.

Furthermore, for CCs, the accuracy measure for a set of N entities can be computed [26]. In contrast with the accuracy measure (i.e., class label set-based evaluation), the F-measure macro averaged over the label-based evaluations for N test entities.

4 Experimental Results

The methodology presented in the previous section is applied to the Darknet dataset, which consists of 158,659 entities. The data are split into training and test data. It can be observed from Table 1 that due to the multiclass problem in the classification stage, the data are split such that class imbalances are accounted for, and datasets have a proportional number of entities. This will ensure that the data splits have similar class distributions of 70% and 30% for training and test data, respectively. In the following subsections, the application and results of the methodology on the detection and classification stages. Furthermore, the Deep Network Classifier is evaluated.

4.1 Detection Stage

The trained deep networks presented in Section 3.2 were applied to the training data portion of the Darknet dataset. Among 200 pre-trained deep networks, the deep network parameters with the tuned values that generate the highest evaluation results are presented in Table 2. The ROC curve plots TPR versus. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both TP and FP. The result of the per-label evaluations and ROC curve for the *detection stage* deep network are presented in Table 4 and Fig. 3, respectively. It can be observed in the evaluation results that the deep network can detect darknet traffic entities in the detection stage with an accuracy of 96.8% and that most of the misdetection entities were normal activity entities detected as darknet traffic. The probability that the model ranks a random positive entity more highly than the AUC can measure a random negative entity. The AUC-ROC is closer to one, which indicates the higher performance of the built Deepnet model in the detection stage. In

cases in which it is desired only to detect darknet activity, completing the detection stage is sufficient to accomplish this task.

Table 4: Per-label evaluation results of the detection and classification stages

Metric	Detection stage	Classification stage
Accuracy	96.8%	94.4%
Precision	94.2%	87.8%
Recall	93.0%	92.2%
F1	0.93	0.89

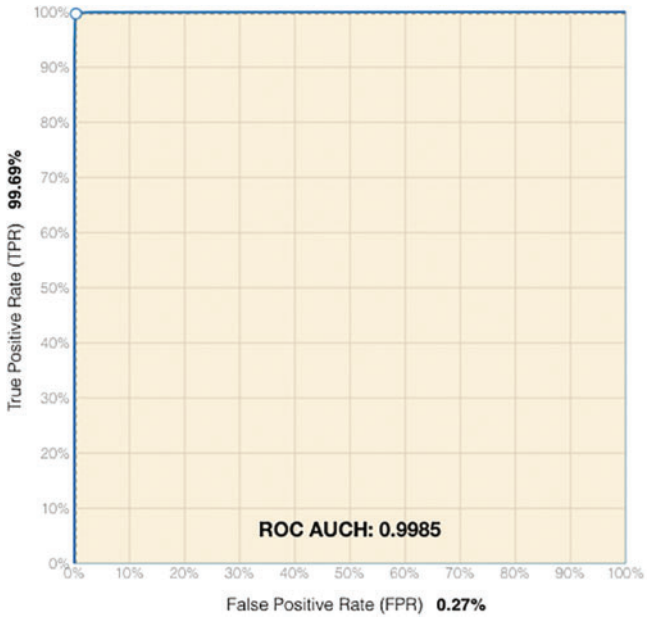

Figure 3: ROC curve for the detection stage deep network

4.2 Classification Stage

To further classify the detected darknet activity, entities classified as darknet entities are passed along the CC to the *classification stage*. Similarly, in the *detection stage*, numerous pre-trained deep networks were applied to darknet-related entities in the Darknet dataset. Furthermore, the deep network's parameters were tuned, and values representing the highest evaluation results are presented in Table 3. The results of the per-label evaluations for entities that have been passed to the *classification stage* deep network are presented in Table 4. It should be noted that the results of the evaluation metrics of Table 4 were based on the overall average for all eight label classes. The accuracy and recall metrics were relatively high. However, the precision value degraded in cases where the model has to deal with several unbalanced class labels; thus, the accuracy metric needs to be more accurate. In situations characterized by poor performance in either precision or recall, the F-measure is a more valid indication of performance. The classification deep network model achieved a score of 89% in the *classification stage*. This is the product of 75.8% and 77% precision scores for detecting

video-streaming and P2P activities, respectively, as few video-streaming entities and P2P activities were classified as audio streaming. One of the reasons for this misclassification is that Audio and Video streaming is commonly used in shared real-time protocols for streaming, such as the Real-time Transport Protocol (RTP) [27]. Accordingly, the features of those class labels tend to have similar observations, primarily when the same RTP is utilized. This only affects the classification of an entity; however, the entity was detected as a darknet activity in the detection stage. In cases in which it is desired to accurately distinguish between P2P, Audio streaming, and Video streaming entities, an additional classifier in the CC can be added; however, this is beyond the scope of this work.

4.3 Deep Network Chain Evaluation

To evaluate the performance of the deep network chain (i.e., label set-based evaluation), the "harsh" loss measure is utilized. This loss measure only considers whether the exact detection and classification labels match. For example, when a darknet activity is accurately detected but misclassified, the loss score considers this a mismatch and is given a zero score. Accordingly, this penalizes the performance evaluation. Furthermore, the F-measure macro averaged over the label set-based evaluations for N test entities is computed. The performance results of the deep network chain were 0.038, 0.96, and 0.91 for loss-measure, accuracy, and F-measure, respectively. In general, classifiers presenting lower loss values and higher accuracy, and F-measure values are considered adequate and can thus be relied upon. Table 5 compares the presented Deep Network Chain with the results of the DeepImage approach [21]. It can be observed from Table 5 that the overall performance of the presented Deep Network Chain outperformed the DeepImage approach. This could be due to the DeepImage method selecting only certain features to build a gray image, then feeding it into a two-dimensional CNN to detect and characterize darknet traffic. However, the superiority of the methodology presented in this work is due to employing two chains of deep networks instead of employing one classifier for detecting and characterizing darknet traffic, where the first deep network detects darknet traffic. The darknet activities are passed into a second classifier for characterization.

Table 5: Comparison with other classifiers of [17]

Approach	Accuracy	F-measure
1D CNN	73%	0.73
DeepImage	86%	0.86
Deep network chain	96%	0.91

5 Conclusions

This paper presented an approach to detecting and classifying darknet traffic by deploying Deep Network Chains. The first classifier in the chain is a deep network binary classifier that detects darknet activities in the *detection stage*. Such activities are passed into the second classifier in the chain. The second classifier is a multiclass deep network that categorizes the hidden services and applications in the darknet *classification stage*. The methodology of this paper was verified on a dataset containing both VPN and Tor traffic. Optimization and parameter tuning were carried out in both stages (i.e., the *detection stage* and the *classification stage*) to achieve more accurate results. To evaluate the performance of the deep network chain, adequate evaluation metrics for classifier chains were utilized, including the loss measure. The performance results of the deep network chain were 0.038, 0.96, and

0.91 for loss measure, accuracy, and F-measure, respectively. It was observed that the misclassification is due to the Audio and Video streaming commonly used in shared real-time protocols. However, in cases in which it is desired to distinguish between such activities accurately, the presented deep chain classifier can accommodate additional classifiers. Furthermore, additional classifiers can be added to the chain to categorize specific activities of interest further.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] K. Demertzis and L. Iliadis, "Evolving smart url filter in a zone-based policy firewall for detecting algorithmically generated malicious domains," *Proceedings of Statistical Learning and Data Sciences*, vol. 9047, pp. 223–233, 2015.

[2] K. Demertzis, L. Iliadis and I. Bougoudis, "Gryphon: A semi-supervised anomaly detection system based on one-class evolving spiking neural network," *Neural Computing and Applications*, vol. 32, no. 9, pp. 4303–4314, 2020.

[3] B. Yang and D. Liu, "Research on network traffic identification based on machine learning and deep packet inspection," in *Proc. of IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conf.*, Chengdu, China, pp. 1887–1891, 2019.

[4] S. Madan, S. Sofat and D. Bansal, "Tools and techniques for collection and analysis of internet-of-things malware: A systematic state-of-art review," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 10, pp. 9867–9888, 2022.

[5] A. Attkan and V. Ranga, "Cyber-physical security for IoT networks: A comprehensive review on traditional, blockchain and artificial intelligence-based key-security," *Complex & Intelligent Systems*, vol. 8, no. 4, pp. 3559–3591, 2022.

[6] M. Abolfathi, I. Shomorony, A. Vahid and J. H. Jafarian, "A game-theoretically optimal defense paradigm against traffic analysis attacks using multipath routing and deception," in *Proc. of the 27th ACM on Symp. on Access Control Models and Technologies*, New York, NY, USA, pp. 67–78, 2022.

[7] B. Xiong, K. Yang, J. Zhao and K. Li, "Robust dynamic network traffic partitioning against malicious attacks," *Journal of Network and Computer Applications*, vol. 87, no. 7, pp. 20–31, 2017.

[8] M. W. A. Nabki, E. Fidalgo, E. Alegre and L. Fernández-Robles, "Torank: Identifying the most influential suspicious domains in the tor network," *Expert Systems with Applications*, vol. 123, no. 1, pp. 212–226, 2019.

[9] H. Bozorgkhou and M. Alimohammadirokni, "Studying and investigating the impact of marketing mix factors on e-purchase via smartphones (case study: Digikala corporation)," *Nexo Scientific Journal*, vol. 35, no. 4, pp. 992–1003, 2022.

[10] P. Tirandazi, A. Rahiminasab and M. J. Ebadi, "An efficient coverage and connectivity algorithm based on mobile robots for wireless sensor networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 23, no. 3, pp. 10, 2022. https://doi.org/10.1007/s12652-021-03597-9

[11] L. Surya, "Machine learning on network security," *International Engineering Journal for Research and Development*, vol. 4, no. 1, pp. 1–4, 2019.

[12] N. Manokaran, V. Varathan and S. Deepak, "Cloud-based big data analytics in smart educational system," in *Deep Learning Innovations and Their Convergence with Big Data*, 1st ed., Pennsylvania, USA: IGI Global, pp. 189–199, 2018.

[13] J. Lam and R. Abbas, "Machine learning based anomaly detection for 5g networks," *arXiv preprint arXiv:2003.03474*, 2020.

[14] Y. Wu, D. Wei and J. Feng, "Network attacks detection methods based on deep learning techniques: A survey," *Security and Communication Networks*, vol. 1, no. 1, pp. 1–17, 2020.

[15] E. Arulprakash and M. Aruldoss, "A study on generic object detection with emphasis on future research directions," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 9, pp. 7347–7365, 2021.

[16] B. Naik, A. Mehta, H. Yagnik and M. Shah, "The impacts of artificial intelligence techniques in augmentation of cybersecurity: A comprehensive review," *Complex & Intelligent Systems*, vol. 8, no. 2, pp. 1763–1780, 2022.

[17] R. Alshammari and A. N. Zincir-Heywood, "Can encrypted traffic be identified without port numbers, ip addresses and payload inspection?," *Computer Networks*, vol. 55, no. 6, pp. 1326–1350, 2011. https://doi.org/10.1016/j.comnet.2010.12.002.355

[18] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun and A. Ghorbani, "Characterization of encrypted and vpn traffic using time-related features," in *Proc. of the 2nd Int. Conf. on Information Systems Security and Privacy*, Rome, Italy, pp. 407–414, 2016.

[19] T. Shapira and Y. Shavitt, "Flowpic: A generic representation for encrypted traffic classification and applications identification," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1218–1232, 2021.

[20] M. H. Al-Mashagbeh and M. Ababneh, "Tor detection using a machine learning approach using correlation-based feature selection with best first and random forest," in *Proc. of the Int. Conf. on Information Technology (ICIT)*, Amman, Jordan, pp. 893–898, 2021.

[21] A. Habibi Lashkari, G. Kaur and A. Rahali, "Didarknet: A contemporary approach to detect and characterize darknet traffic using deep image learning," in *Proc. of the 10th Int. Conf. on Communication and Network Security (ICCNS)*, New York, NY, USA, pp. 1–13, 2020. https://doi.org/10.1145/3442520.3442521

[22] J. Read, B. Pfahringer, G. Holmes and E. Frank, "Classifier chains for multilabel classification, machine learning, machine learning and knowledge discovery in databases," in *Proc. of the Joint European Conf. on Machine Learning and Knowledge Discovery in Databases*, Bled, Slovenia, vol. 5782, pp. 333–359, 2009.

[23] A. H. Lashkari, G. Draper-Gil, M. Mamun and A. Ghorbani, "Characterization of tor traffic using time-based features," in *Proc. of the Int. Conf. on Information Systems Security and Privacy*, Porto, Portugal, pp. 253–262, 2017.

[24] S. Niu, Y. Liu, J. Wang and H. Song, "A decade survey of transfer learning (2010–2020)," *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 2, pp. 151–166, 2020.

[25] K. Dembczynski, W. Waegeman, W. Cheng and E. Hullermeier, "On label dependence and loss minimization in multilabel classification," *Machine Learning*, vol. 88, no. 1–2, pp. 5–45, 2012.

[26] S. Godbole and S. Sarawagi, "Discriminative methods for multi-labeled classification," in *Proc. of the 8th Pacific-Asia Conf. on Advances in Knowledge Discovery and Data Mining*, Sydney, Australia, vol. 3056, pp. 22–30, 2004. https://doi.org/10.1007/978-3-540-24775-3_5

[27] T. -Y. Hsiao and S. -M. Yuan, "Practical middleware for massively multiplayer online games," *IEEE Internet Computing*, vol. 9, no. 5, pp. 47–54, 2004.