Dijlah Journal of Engineering Sciences (DJES) **Vol. 1, No. 3, December, 2024, pp. 119-130**
ISSN: Printed: 3078-9656, Online: 3078-9664, paper ID: 21

Detecting malicious data using deep learning

Hashim Dia'a Hashim Al Noimy[1], Tiba Dia'a Hashim Al Noimy[2]
University of Diyala, College of Engineering, Computer Department
tebaalnoimy@gmail.com, hashim1997h3h13@gmail.com

Article Info

Article history:

Received Nov. 30,2024
Revised Dec. 15, 2024
Accepted Dec. 22, 2024

Keywords:
traffic ,
CNN ,
Benign ,
Malware

ABSTRACT

Traffic classification is a fundamental task in network anomaly detection and intrusion prevention systems. By accurately identifying the types of traffic traversing a network, security professionals can detect and mitigate various threats, such as malicious attacks, unauthorized access, and network congestion. Traditional methods of traffic classification often rely on handcrafted features, which can be time-consuming and prone to errors.

In this research, we present a novel approach that leverages artificial intelligence to streamline and improve the process of traffic classification. Specifically, we propose a convolutional neural network (CNN) model that directly processes raw traffic data as images. This eliminates the need for manual feature extraction, which can be a laborious and error-prone task.

Our CNN model is designed to capture the underlying patterns and characteristics of network traffic. By processing raw traffic data as images, the model can learn to identify distinctive features that differentiate various traffic types.

Corresponding Author:

Tiba Dia'a Hashim Al Noimy

tebaalnoimy@gmail.com

1. INTRODUCTION

Traffic classification is a critical component of network management, especially in the realm of network security. It involves categorizing network traffic according to the specific applications that generated it. This classification serves as a foundational step in activities such as anomaly detection, which helps identify potential security breaches and unauthorized usage of network resources.[1]

Traffic classification primarily employs four techniques:[1] port-based, deep packet inspection (DPI)-based, statistical-based, and behavioral-based. Traditional methods like port-based and DPI-based rely on predefined rules to categorize traffic. In contrast, statistic-base and behavior-base approaches utilize ML to classifying traffics via analyzing pattern within experiential data by use selected features.[2]

While traditional machine learning methods offer advantages over rule-based approaches in handling encrypted traffic and reducing computational costs, they present a new challenge: the need to carefully design appropriate features. Recent research has extensively focused on addressing this issue[3].

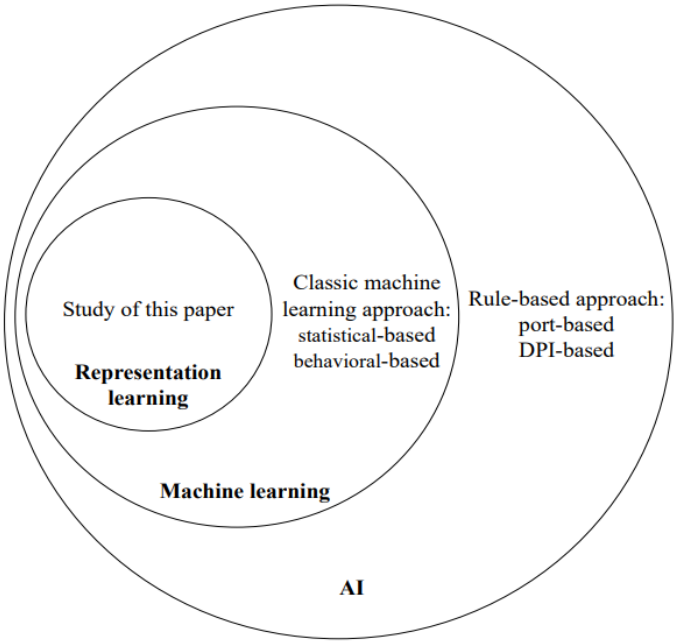

Figure 1: Traffics classifications taxonomy in an AI perspective

Representation learning, a burgeoning field within machine learning, has garnered substantial attention in recent years. This approach involves automatically extracting meaningful features from raw data, eliminating the laborious task of manual feature engineering[4] . Deep learning, a prominent subset of representation learning, has exhibited exceptional performance across diverse domains, including image classification and speech recognition.[5] [6] .

In this research, our primary goal is to investigate the potential of representation learning in accurately classifying malicious network traffic. Figure 1 outlines the classification of traffics classifications from an AI standpoint, while Figure 2 visually compares the various workflows of these approaches, highlighting the components capable of learning from data[2].

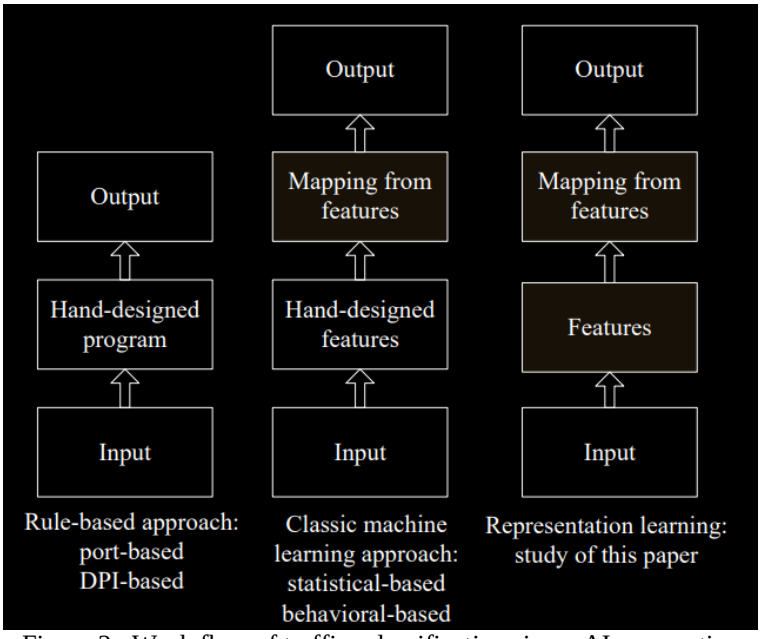

Figure 2: Work flow of traffics classifications in an AI perspective

This research employed a convolutional neural network (CNN), a widely used technique for learning representations. Instead of manually extracting features from traffic data, the raw data was directly treated as images and classified using the CNN, a common task for this type of neural network [7]. This research pioneered the applications of demonstration learning to classify malwares traffics by use rows traffics data. By treating traffic data as images and employing a convolutional neural network (CNN), we successfully categorized malicious traffic. To address the disparity between continuous traffic data and discrete image data, we explored various traffic representations and experimentally determined the most effective approach.

The paper structured is as follow:

- Section 2: presents a comprehensive overview of existing study in the fields and explains the rationale behind our proposed approach.
- Section 3: Details the methodology of our convolutional neural network (CNN) model, outlining the steps involved in traffic classification.
- Section 4: Presents the result and examination of our experiments, highlighting the performance and accuracy of our model.
- Section 5: Discusses the limitations of our current approach and identifies potential areas for future research and improvement.
- Section 6: Offers concluding remarks, summarizing our findings and contributions to the field.

2. RELATED WORK

The field of traffic classification has witnessed the development of mature rule-based approaches. Prior research has primarily concentrated on refining map rule and optimizing performances. The authors in [8] offer a thorough overview of the prevalent DPI-based methods for traffic classification.

Academic research has extensively explored the application of classical machine learning techniques to traffic classification, with a particular emphasis on optimizing feature selection. Dhote et al. [3] conducted a comprehensive survey of various techniques employed in feature selection for internet traffic classification.

Existing research on traffic classification using representation learning is limited. While Gao et al. [9] and Javaid et al. [10] explored the application of deep belief networks and sparse autoencoders for malware traffic classification, both studies relied on handcrafted flow feature datasets like KDD CUP1999 and NSL-KDD.

Inspired by these previous studies, our research aims to classify malware traffic directly from raw traffic data. We employ a convolutional neural network (CNN) as our chosen representation learning method.

3. METHODOLOGY

As noted by Dainotti et al. [13] , a significant obstacle hindering progress in traffic classification research is the scarcity of diverse and publicly accessible traffic trace datasets. Many studies investigating malware traffic classification rely on proprietary or self-collected traffic data, which can compromise the generalizability of their findings. Traditional ML approach predominantly effort on features selections methods, leading to publicly available datasets that primarily consist of pre-defined flow features rather than raw traffic data. Notable examples of such datasets include KDD CUP1999 and NSL-KDD, which offer a fixed set of 41 features.

Unfortunately, these datasets are insufficient for analyzing raw network traffic. There are only a few datasets that provide raw traffic data with enough samples of both normal and malicious activity.

To address these challenges, the USTC-TFC2016 dataset was created. This dataset is divided into two main fragments, as shown in Tables 1 and Table 2. Fragment 1 covers ten different kinds of malwares traffics composed from community websites. This sample was gathered from practical networks environments thru CTUs researchers among 2011 to 2015. Large traffics samples were shortened, while smaller ones were combined if they originated from similar applications. Fragment 2 includes 10-categories of usual traffics composed by use the IXIA BPS [16], qualified networks traffics simulations device. For more details on the simulation process, please visit the IXIA BPS website. To cover a variety of traffic types, the dataset includes ten different kinds of traffic representing eight common application classes. The USTC-TFC2016 dataset is 3.71GB in size and is stored in the pcap format.

Table 1: USTC-TFC2016 PART 1 (MALWARES TRAFFICS)

Name	CTU num	Binary MD5	process
Cridex	108-1	25b8631afeea279ac00b2da70fffe18a	original
Geodo	119-2	306573e52008779a0801a25fafb18101	part
Htbot	110-1	e515267ba19417974a63b51e4f7dd9e9	original
Miuref	127-1	a41d395286deb113e17bd3f4b69ec182	original
Neris	42,43	bf08e6b02e00d2bc6dd493e93e69872f	merged
Nsis-ay	53	eaf85db9898d3c9101fd5fcfa4ac80e4	original
Shifu	142-1	b9bc3f1b2aace824482c10ffa422f78b	part
Tinba	150-1	e9718e38e35ca31c6bc0281cb4ecfae8	part
Virut	54	85f9a5247afbe51e64794193f1dd72eb	original
Zeus	116-2	8df6603d7cbc2fd5862b14377582d46	original

Table 2: USTC-TFC2016 PART 2 (NORMALS TRAFFICS)

Name	Class	Name	Class
BitTorrent	P2P	Outlook	Email/WebMail
Facetime	Voice\Video	Skype	Chat/IM
FTP	Data Transfer	SMB	Data Transfer
Gmail	Email/WebMail	Weibo	Social NetWork
MySQL	Database	WorldOfWarcraft	Game

A. Representation of Networks Traffics

In the initial stages of our ML-base traffics classifications method, it's necessary to divide continuous traffics into distinct unit depend on a specific level of granularity. Additionally, each packet can be selected from different OSI or TCP/IP layers. The following section introduces the process of selecting traffics granularities and packets layer within suggested method.

- Granularity of Traffics

Networks traffics can be divided into different units based on various levels of granularity, such as TCPs connections, flows, sessions, services, and hosts [13] . Diverse granularities levels result in distinct traffics unit. In our approach, we use flows and sessions as the chosen granularity, following the practices of many researchers in the field. A flow is defined as a group of packets with the same 5-tuple, including the source IP, source port, destination IP, destination port, and transport-level protocol. A session includes bidirectional flows, capturing traffic in both directions. The formal description of this distinction is as follows:

- Raw Traffics: Wholly packet is define as sets of $= \{p^1, \dots, p^{|p|}\}$, with each packets are define as $p^i = (x^i, b^i, t^i)$, $i = 1,2, \dots, |P|$. A first component x^i stand to 5-tuples, a second component stand to the size of packets $b^i \in [0, \infty\}$ in byte, and the latest component stand to started time of transmissions $t^i \in [0, \infty\}$ in second.
- Flows: is the sets of rows traffics P which could be divide to multiple subsets. Wholly packet in subsets is arrange in times orders, i.e. $\{p^1 = (x^1, b^1, t^1), \dots, p^n = (x^n, b^n, t^n)\}$, $t^1 < t^2 < \cdots < t^n$. The subsets are define as a flows $f = (x, b, d, t)$. A first component is similar as 5-tuples , i.e. $x = x_1 = \cdots = x_n$. The second component is the total of sizes of entirely packet in flows. The third component is the flows period $d_1 = t^n - t^1$. A latest component is a started times of transmissions of first packets. A complete rows traffics could be converts to flow $F = \{f^1, \dots, f^n\}$.
- Sessions: the sessions include all direction of flow, i.e. the sources and destinations IP/ports are substitutable.

Numerous flow or session might have several sizes, nonetheless the inputs data sizes of CNNs need to be unvarying. Therefore, just the 1st n byte (n = 784) of every flows or sessions is use. This choice could be intuitively explained. Generally, the noticeable part of a flow or session typically contains connections data with fewer contents data, which must be reflects the inherent characteristic of flows or sessions. These select align with other approaches such as [18, 19], that investigated malwares traffics identifications by use classical ML approaches. Additionally, by

using only the first few hundred bytes, this technique could be higher lightweight than numerous rules-based approaches.

- Packets layer

When analyzing packet layers, it's generally expected that the inherent characteristics of network traffic would be evident in the applications layers of the TCP/IPs models, specifically layers seven of OSI models. As instance, protocols like SMTP are associated with email traffic, while HTTP is linked to browsers traffics. Depend on these assumptions, the authors in [12] focuses exclusively on layers 7, referring to it as TCPs sessions payloads. However, it's important to consider that data from other layers can also provide valuable traffics features data. As instance, ports data in the transporting layers could recognize definite application or services.
Most applications use standard port numbers, and certain flags data could help identifying networks attacks such as SYN attacks and RSTs attacks. Hence, we considered two options for packet layer selection: including entirely layer and individual considering layer seven (L7). We can note that including IP and MACs data in sessions or flows can potentially interfere with the feature extraction process.
To address this issue, it's necessary to remove such information using randomization techniques, often referred to as traffics sanitizations. The study examines 4 different traffics representations types: Flows + All, Flows + L7, Sessions + All, and Sessions + L7. These representations were evaluated by testing their performance using the two traffic datasets introduced in Part A. Ultimately, the study identifies the most effective representation type based on the results obtained from eight experiments.

- Raw Traffics: Wholly packet is define as sets of $= \{p^1, \dots., p^{|p|}\}$, with each packets are define as $p^i = (x^i, b^i, t^i)$, $i = 1,2, \dots., |P|$. A first component x^i stand to 5-tuples, a second component stand to the size of packets $b^i \in [0, \infty\}$ in byte, and the latest component stand to started time of transmissions $t^i \in [0, \infty\}$ in second.
- Flows: is the sets of rows traffics P which could be divide to multiple subsets. Wholly packet in subsets is arrange in times orders, i.e. $\{p^1 = (x^1, b^1, t^1), \dots., p^n = (x^n, b^n, t^n)\}$, $t^1 < t^2 < \cdots < t^n$. The subsets are define as a flows $f = (x, b, d, t)$. A first component is similar as 5-tuples , i.e. $x = x_1 = \cdots = x_n$. The second component is the total of sizes of entirely packet in flows. The third component is the flows period $d_1 = t^n - t^1$. A latest component is a started times of transmissions of first packets. A complete rows traffics could be converts to flow $F = \{f^1, \dots., f^n\}$.
- Sessions: the sessions include all direction of flow, i.e. the sources and destinations IP/ports are substitutable.

Numerous flow or session might have several sizes, nonetheless the inputs data sizes of CNNs need to be unvarying. Therefore, just the 1st n byte (n = 784) of every flows or sessions is use. This choice could be intuitively explained. Generally, the noticeable part of a flow or session typically contains connections data with fewer contents data, which must be reflects the inherent characteristic of flows or sessions. These select align with other approaches such as [18, 19], that investigated malwares traffics identifications by use classical ML approaches. Additionally, by using only the first few hundred bytes, this technique could be higher lightweight than numerous rules-based approaches.

- Packets layer

When analyzing packet layers, it's generally expected that the inherent characteristics of network traffic would be evident in the applications layers of the TCP/IPs models, specifically layers seven of OSI models. As instance, protocols like SMTP are associated with email traffic, while HTTP is linked to browsers traffics. Depend on these assumptions, the authors in [12] focuses exclusively on layers 7, referring to it as TCPs sessions payloads. However, it's important to consider that data from other layers can also provide valuable traffics features data. As instance, ports data in the transporting layers could recognize definite application or services.
Most applications use standard port numbers, and certain flags data could help identifying networks attacks such as SYN attacks and RSTs attacks. Hence, we considered two options for packet layer selection: including entirely layer and individual considering layer seven (L7). We can note that including IP and MACs data in sessions or flows can potentially interfere with the feature extraction process.
To address this issue, it's necessary to remove such information using randomization techniques, often referred to as traffics sanitizations. The study examines 4 different traffics representations types: Flows + All, Flows + L7,

Sessions + All, and Sessions + L7. These representations were evaluated by testing their performance using the two traffic datasets introduced in Part A. Ultimately, the study identifies the most effective representation type based on the results obtained from eight experiments.

- Preprocess of data

The preprocessing of data includes transform raw traffics data (in pcap format) into a suitable format for CNN input. This process consists of four steps: traffic splitting, traffic cleaning, image generation, and data augmentation. To facilitate these steps, a dedicated toolkit called USTC-TL2016 was developed. The overall data preprocessing workflow is illustrated in Figure 3.

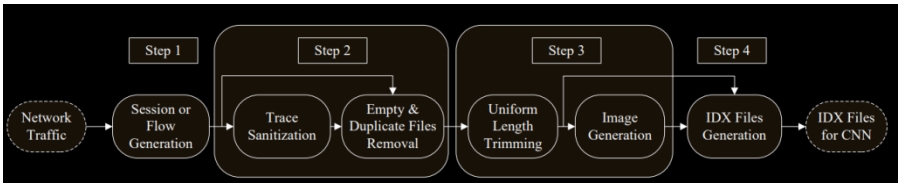

Figure 3: Preprocessing of data

1- Step 1: Traffic Splitting:

Traffic splitting involves dividing continuous rows traffics into multiple discrete traffics unit. The inputs data formatting is in (pcap). For representations type such as Flows + All or Sessions + All, the outputs data formatting remains as (pcap). However, for representations types like Flows + L7 or Sessions + L7, the outputs data formatting is in bin format.

2- Step 2: Traffics Cleaning:

Traffic cleaning involves two main actions. First, it includes the option to perform traffics anonymization/sanitizations that randomize the MACs addresses in the data links layers and the IP address in the IP layer. This step is not always necessary, such as when all traffic originates from the same network, where MAC and IP addresses may no longer be distinguishing factors. In such cases, this action can be skipped [9].

The 2nd actions in traffics cleaning are file cleaning. Certain packet may not has an applications layers, resulting in empty bin files. Additionally, identical content in packets can lead to the generation of duplicate files, which can introduce bias during CNN training. To address this, empty and duplicate files are removed. The data format remains unchanged in this step.

3- Step 3 : Image Generation:

Image generation involves two key steps. First, all files are trimmed to a uniform length. If the file is exceed 784 byte, which truncated to 784 byte. When the files are smaller than 784 byte, it is padded with 0x00 at the end to reach 784 byte. Then, the resulting file of the equal sizes is converts into gray scales image. Every byte in the origin files corresponds to pixels, where 0x00 represents blacks and 0xff represent whites.

The USTC-TFC2016 traffics datasets were processing by use the USTC-TK2016 toolkits, resulting in a total of 752,040 records. Table 3 displays the result. Since session contains bidirectional flow, the numbers of session is typically lower than the number of flow.

Table 3: SESSIONS & FLOW COUNT RESULTS

Dataset	Representation	Count Range	Count Total
Malware	Flow+ALL	6000~17178	134563
	Flow+L7	4569~13968	76716
	Session+ALL	6000~8629	71008
	Session+L7	4569~7592	63120
	Total	----------	406633
Benign	Flow+ALL	10051~17008	138145
	Flow+L7	8908~16391	126665
	Session+ALL	5134~9634	71692
	Session+L7	4952~9476	70131
	Total	----------	345407
Total	-----	------	752040

4. VISUALIZATION ANALYSIS

In this section, we analyze the images generated during the third step of the data preprocessing procedure. Each grayscale image is 784 bytes (28x28 pixels). The visualization results of the Session + All representation can be seen in Figures 2 and Figure 3. The results of another 3-representation type are in general same to these.

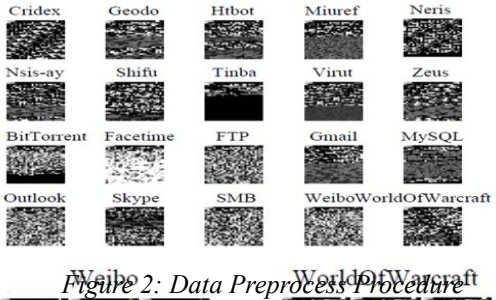

Figure 2: Data Preprocess Procedure

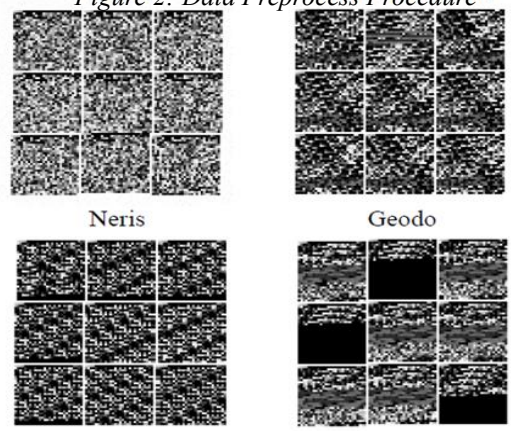

Figure 3: Visualization of All Classes of Traffic

Figure 2 and Figure 3 shows the visualization results for all classes, clearly demonstrating their distinctness. While most images show significant differences, a few have similarities, such as between FTP and SMB. Figure 3 highlights the consistency within each traffics classes. The 9- random chosen image from a single session and four randomly chosen classes are displayed. Interestingly, the images from Weibo, WOW, and Neris have similar textures. Additionally, the Geodo class can be divided into two subclasses, with images in each subclass still showing significant similarities. Overall, the remaining sixteen classes exhibit a generally consistent pattern. Based on our visualization analysis, we can conclude that different traffic classes are distinguishable, and each class has a high level of consistency. Therefore, we expect our approach to perform well.[14]

B. CNN Architecture

The CNN images analyzing processing start by read traffics images in dimensions of 128x128x3, which is extracts in the 3rd step of data preprocessing. The pixel values in images are then normalized to the range of [0, 1] from [0, 255]. Next, the 1st convolutional layers, C1, apply a convolutional operation to images by use 32 kernel and sized with 3x3. The generated 32 features map, each sized of 128x128. A 2x2 pool processing is then apply to feature map created by the C1 layers.

Next, two more convolution layers, C3 and C2, are used, each with the similar kernels sizes as C1 but within 64 channel. These creates 64 distinct map, each within a sizes of 64x64. Then, 64 distinct map within a sized of 32x32 are generated, follow by 3-layer, C3, C4, and C5, within similar kernels sizes as before nonetheless with 128 channel. These processing results in the production of 128 distinct maps, each sized 16x16.

In order to compute the probabilities of every classes, a sigmoid function was use, and dropout is implemented to prevent overfitting. This CNN structure is used in the classifiers presented in this study.[19]

C. Scalability Study

The proposed method was applied to two different scenarios using two kinds of CNNs classifier include binary classifiers and an 8-class classifiers. In 1st situation, the data was classify to two categories: malicious data and normal data, representing a binary classifications task. In 2nd scenario, the output from the binary classification was used, and a subsequent classification was performed with 8 classes to identify each traffic class sequentially.[22]

5. EVALUATION

- EVALUATION METRICS

Four evaluation metrics were employed to assess the performance of the classifiers: accuracies (A), precisions (P), recalls (R), and F1-scores (F1). Accuracies were use to measure the complete performances of classifiers. Precisions, recalls, and F1-scores are use to calculate the performances of each individual traffic classes.

$$A = \frac{TP+TN}{TP+FP+TN+FN} \quad \text{.....(1)}$$

$$P = \frac{TP}{TP+FP} \quad \text{.....(2)}$$

$$R = \frac{TP}{TP+TN} \quad \text{.....(3)}$$

$$F1 = \frac{2PR}{P+R} \quad \text{.....(4)}$$

In the evaluation process, the following terms are used: TP (true positives) represents the number of instances correctly classified as X, TN (true negatives) represents the number of instances correctly classified as Not-X, FP (false positives) represents the number of instances incorrectly classified as X, and FN (false negatives) represents the number of instances incorrectly classified as Not-X.[16]

- Representation Experiment Results and Analysis

A confusion matrix is a table that shows the difference between the actual and predicted outcomes of a machine learning model. In this case, the model is a web security system trained to detect web attacks. The matrix consists of six squares, each with a different color. Each square represents the proportion of correct or incorrect predictions for a specific case. For example, the top-left square represents the proportion of correct predictions that a user is not wearing a mask.

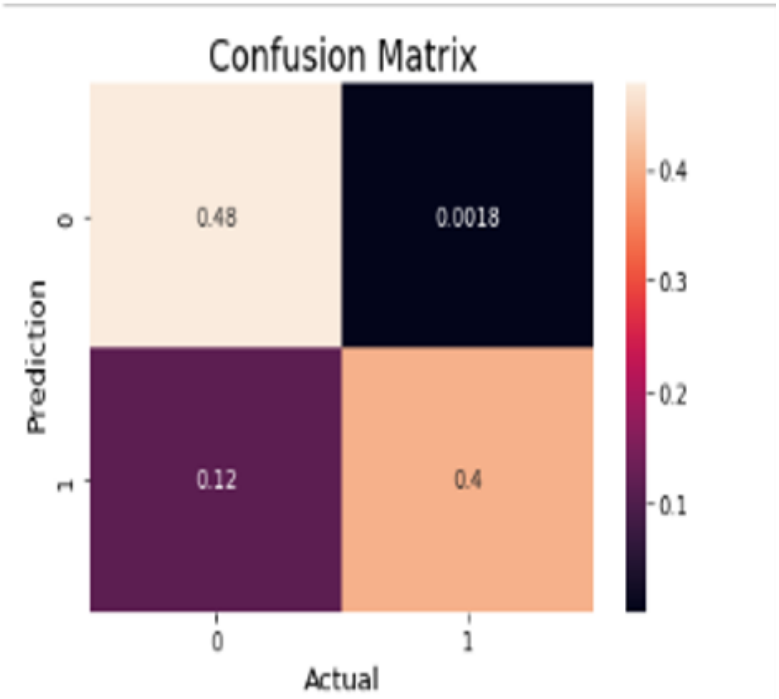

Figure 4: confusion matrix for the binary classification

A confusion matrix is a table that summarizes the performance of a classification model. It shows the number of correct and incorrect predictions made by the model for each class. In this case, the confusion matrix has two rows and two columns, corresponding to the two classes, Benign and Malware.

The rows of the confusion matrix represent the actual class of the data, and the columns represent the predicted class of the data. The cells in the matrix show the number of data points that were classified correctly or incorrectly.

Here is an explanation of the results shown in the confusion matrix :

1) True positives (TP): The number of data points that were correctly classified as Malware. In this case, there are 48 TP.
2) True negatives (TN): The number of data points that were correctly classified as Benign. In this case, there are 40 TN.
3) False positives (FP): The number of data points that were incorrectly classified as Malware. In this case, there are 2 FP.
4) False negatives (FN): The number of data points that were incorrectly classified as Benign. In this case, there are 10 FN.

From the confusion matrix, we can calculate the following performance metrics:

	precision	recall	f1-score	support
0	0.80	1.00	0.89	1047
1	1.00	0.78	0.87	1139
accuracy			0.88	2186
macro avg	0.90	0.89	0.88	2186
weighted avg	0.90	0.88	0.88	2186

Figure 5: Binary classifier results

Based on these performance metrics, we can conclude that the model is performing well in classifying data points as Benign or Malware. The accuracy, precision, recall, and F1-score are all high.

5. CONCLUSIONS

Traffic classification is the initial step in detecting network anomalies or network-based intrusion detection systems and plays a significant role in network security. We propose a method for classifying malicious traffic using convolutional neural networks by treating traffic data as images. This approach eliminates the need for manually engineered features, instead directly taking raw traffic data as input to the classifier. The method was validated using a binary classification scenario, and experimental results demonstrate that our proposed approach can meet the accuracy requirements for practical applications Where the data was classified correctly with 88% accuracy.

References

[1] E. Biersack, C. Callegari and M. Matijasevic, Data traffic monitoring and analysis. Berlin: Springer, 2013.

[2] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning, Book in preparation for MIT Press, 2016.

[3] Y. Dhote, S. Agrawal. "A Survey on Feature Selection Techniques for Internet Traffic Classification". in Computational Intelligence and Communication Networks, Jabalpur, 2015, pp. 1375-1380.

[4] Y. Bengio, A. Courville and P. Vincent, "Representation learning: A review and new perspectives", IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, pp. 1798-1828, Aug. 2013.

[5] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning", Nature, vol. 521, pp. 436-444, May 2015.

[6] Z. Qingqing, L. Yong, W. Zhichao, P. Jielin and Y. Yonghong, "The Application of Convolutional Neural Network in Speech Recognition", Microcomputer Applications, vol. 3, pp. 39-42, June. 2014.

[7] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy and B. Shuai, "Recent Advances in Convolutional Neural Networks", arXiv preprint arXiv:1512.07108, 2015.

[8] M. Finsterbusch, C. Richter, E. Rocha, J. A. Muller and K. Hanssgen, "A Survey of Payload-Based Traffic Classification Approaches", Communications Surveys & Tutorials IEEE, vol. 16, no. 2, pp. 1135- 1156, 2014.

[9] N Gao, L Gao and Q Gao, "An Intrusion Detection Model Based on Deep Belief Networks", Advanced Cloud and Big Data (CBD) 2014 Second International Conference on, pp. 247-252.

[10] A. Javaid, Q. Niyaz, W. Sun and M. Alam. "A Deep Learning Approach for Network Intrusion Detection System." in Proc.9th EAI International Conference on Bio-inspired Information and Communications Technologies. New York, 2016.

[11] M. Tavallaee, E. Bagheri, W. Lu and A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set", Proc. 2009 IEEE Int. Conf. Comput. Intell. Security Defense Appl., pp. 53-58.

[12] Z. Wang, "The Applications of Deep Learning on Traffic Identification." https://goo.gl/WouIM6

[13] A. Dainotti, A. Pescape and K. Claffy, "Issues and future directions in traffic classification", Network IEEE, vol. 26, no. 1, pp. 35-40, 2012.

[14] G. Creech and J. Hu, "Generation of a new ids test dataset: Time to retire the kdd collection", Wireless Communications and Networking Conference (WCNC) 2013 IEEE, pp. 4487-4492, 2013.

[15] F. Haddadi and A. Nur Zincir-Heywood, “Data confirmation for botnet traffic analysis,” in Proc. 7th Int. Symp. FPS, to be published.

[16] CTU University, The Stratosphere IPS Project Dataset, https://stratosphereips.org/category/dataset.html, 2016.

[17] Ixia Corporation, Ixia Breakpoint Overview and Specifications, https://www.ixiacom.com/products/breakingpoint, 2016.

[18] Z. B. Celik, R. J. Walls, P. McDaniel and A. Swami, "Malware traffic detection using tamper resistant features," Military Communications Conference, MILCOM 2015 - 2015 IEEE, Tampa, FL, 2015, pp. 330- 335.

[19] W. Li, "Efficient Application Identification and the Temporal and Spatial Stability of Classification Schema", Computer Networks, vol. 53, pp. 790-809, Apr. 2009.

[20] D. Koukis, S. Antonatos, D. Antoniades, E. P. Markatos and P. Trimintzios, "A Generic Anonymization Framework for Network Traffic," 2006 IEEE International Conference on Communications, Istanbul, 2006, pp. 2302-2309.

[21] A. W. Moore, D. Zuev, and M. Crogan. Discriminators for use in flowbased classification. Technical Report RR-05-13, Department of Computer Science, Queen Mary, University of London, September 2005.

[22] MH. Bhuyan, DK. Bhattacharyya and JK. Kalita, "Network Anomaly Detection: Methods, Systems and Tools," in IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 303-336, First Quarter 2014