ARIEL UNIVERSITY

MASTER THESIS PROPOSAL

Learning Approaches for Robust Classification of Operation system, Browser and Application on Encrypted Traffic

Author:
Levi Maatuk OZ

Supervisor:
Dr. Dvir AMIT

Department of Computer Science

February 3, 2021

Declaration of Authorship

I, Levi Maatuk Oz, hereby declare that this thesis proposal entitled, "Learning Approaches for Robust Classification of Operation system, Browser and Application on Encrypted Traffic" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Ariel University

Abstract

Faculty of Natural Sciences
Department of Computer Science

Master

Learning Approaches for Robust Classification of Operation system, Browser and Application on Encrypted Traffic

by Levi Maatuk Oz

TODO

Keywords: operation-system, browser-program, internet-application, internet-network, internet-communication, encryption-protocols, machine-learning, deep-learning, adversarial-opponent, robust learning model

Contents

List of Figures

List of Tables

List of Abbreviations

BOA	**B**rowser-Program, **O**peration-System and Internet **A**pplication
OS	**O**peration **S**system
AI	**A**rtificial **I**ntelligence
ML	**M**achine **L**earning
DL	**D**eep **L**earning
AO	**A**dversarial **O**pponent
RF	**R**andom **F**orest
ET	**E**extra **T**rees
SVM	**S**upport **V**ector **M**achine
GBM	**G**radient **B**oosting **M**achine
XGBOOST	e**X**tream **G**radient **B**oosting **M**achine
LIGHTGBM	**L**ight **G**radient **B**oosting **M**achine
SSL	**S**ecure **S**ocket **L**ayer
TLS	**T**ransport **L**ayer **S**ecurity
HTTP	**H**ypter **T**ext **T**ransfer **P**rotocol
HTTPS	**H**ypter **T**ext **T**ransfer **P**rotocol **S**ecure
VPN	**V**irtual **P**rivate **N**etwork
DPI	**D**eep **P**acket **I**nspection
ISP	**I**nternet **S**ervice **P**rovider
PSD	**P**ayload **S**ize **D**istribution
NAT	**N**etwork **A**ddress **T**ranslation
QoS	**Q**uality **o**f **S**ervice
QoE	**Q**uality **o**f **E**xperience
TP	**T**rue **P**ositive
FN	**F**alse **N**egative
FP	**F**alse **P**ositive
TN	**T**true **N**egative
NN	**N**eural **N**etwork
CNN	**C**onvolutional **N**eural **N**etwork
DoS	**D**enial **o**f **S**ervice

Chapter 1

Introduction

Encrypted Traffic Classification [Fig. 1.1] as a sub-domain in network traffic analytics, aims to identify information about the transferred data or the user in the network, usually over the internet. The value of traffic analyzing over the internet results to be precious, when network-traffic has been early explored for user-tracking, Website-usability, Quality-of-Service (QoS), Quality-of-Experience (QoE), and internet-security aspects. QoE tasks like network-management, user management, bandwidth control, user fairness, etc. became conceivable operations, while network performance challenges such as stability, availability, reliability, scalability of internet-services, contemplated providing appropriate QoS.

The importance of traffic analysis was also raised in security manners, where the ability to identifying illegitimate messages improved intrusion detection, and anomaly detection tasks contributed to overthrow malicious traffic. On the other hand, the use of the internet can put any user at risk of a Passive cyber-attack, where the hostile entity in the network collects the information transferred by some other honest user in the network, and makes use of it, without affecting the system resources.

Identifying user traffic usually requires performing a deep packet inspection (DPI), that decodes the traffic and inspects the header or the payload. Artificial-Intelligence (AI) including big data and learning algorithms, in addition to DPI, improves the abilities for analyzing encrypted traffic.

1.1 Problem of Interest

Our proposal will introduce the ability for robust classification of the BOA tuple (Operation-system, Browser-software, internet-Application) used by some entity over the encrypted internet-network. BOA classification has been examined [29] and presented 96% success using the supervised learning model Random Forest (RF). Traffic classification using AI models develops both academia and industry, required to examine the robustness of the models in Adversarial-Opponent (AO) cases. When considering the AI classification task, AO usually represented as a user trying to interrupt the identification process.

1.2 Our Contribution

Our work will survey several state-of-art learning models of two AI approaches, to specify the proper learning algorithm for robust classification of the BOA. The robustness property will be examined by measuring the performance of different models facing several AO cases, like manipulate featured data values, using padding

techniques, modify cipher-suites, communicating through VPN, and Tor. Examining the relationship between classes, studying the ML performance, and select features that are uniquely and highly representative for the task [47]. The robustness of different learning models will be presented with appropriate features sets.

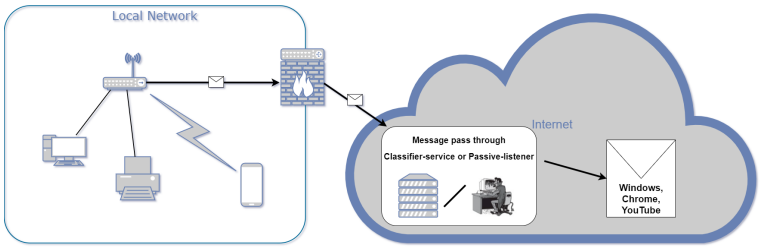

FIGURE 1.1: Internet-Network Traffic Classification

Chapter 2

Background

2.1 Monitoring Network Traffic

Any activity over the internet-network requires transmitting data, includes personal user information, over public channels, through the network. Such information was discovered to be valuable in several manners. This can expose the honest user to passive hackers' attacks. A passive cyber-attack is an attempt of a hostile entity in the network to collect and make use of information transferred by some other honest user in the network and but does not affect system resources. It occurs when an using the internet network and meanwhile attacker is listening to the traffic over the network [FIGURE1.2 **??**]. Usually, the passive cyber-attack includes eavesdropping on or monitoring the transmission, while the goal of the opponent is to obtain information that is being transmitted [48]. The first known simplest method for monitoring traffic is the Port-based classification for network traffic identification. This technique becomes out-of-date because of network address translation (NAT), port forwarding, protocol embedding, and random ports assignments which are common in model internet-communication.

Next, the deep packet inspection (DPI) technique which was first used in The Advanced Research Projects Agency Network (ARPANET) were they examined a wide-area packet-switching network with distributed control and was one of the first networks to implementing the TCP/IP protocol suite. This technique decode network-traffic and look at the contents or payload of that traffic, using predefined patterns like regular expressions as signatures for protocol parameters [56]. It inspects the packet's headers and data content of the packet, evaluating conclusions by examines by Internet-network packet parameters **??**]. The abilities of the technique made it possible to the performing DPI for identifying user activity on the internet-network [9]. Some disadvantages of using DPI can come up when updates in protocols policies or application communication systems in addition to encrypted communication which is very hard to inspect. DPI also found to be useful in QoS operations and carry lot of restricts due to privacy policies.

At last, the statistical and natural classification results as proper technique for identifying network packets [5]. Usually, those methods require extracting features like packet-size, packet-time-arrival and applying supervised learning algorithms for classification. Identifying network packets can be done by using each packet as sample for learning (packet-based), or mostly common, as collection of related packets as session (session-based) for representing sample. A hybrid approach that uses DPI for extracting features about the traffic for machine learning results as the most efficient technique for the task. Naturally, after hitching machine learning methods to the problem, deep learning was also raised efficiently and been used in several ways for classifying network traffic and packets [57]. Those will present in detail later.

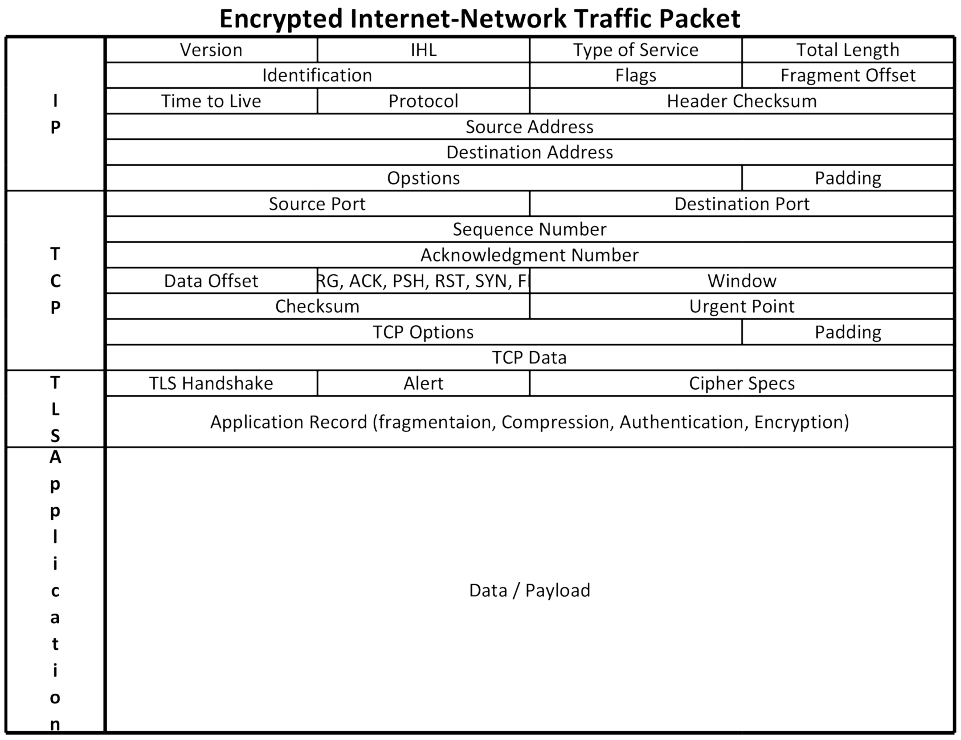

FIGURE 2.1: Encrypted Internet-Network Traffic Packet Structure and Parameters

2.2 Encryption

Encryption started in ancient spartan times, Spartans have used a wooden rod to wrap it with leather contains a message, only with the original rod the message could be read correctly. Later, in Rome, Julius Caesar invented a substitution cipher that shifts characters by three places. It continues with Giovan Battista Bellaso who presented the first cipher to use a proper encryption key and Charles Wheatstone who invented the Play-fair cipher, which encrypts pairs of letters.

Digital encryption started when Edward Hebern invented the first of the rotor machine which uses an embedded key in a rotating and encodes a substitution table that is changed every new input. German engineer Arthur Scherbius developed the Enigma machine that uses several rotors. In war-world two, Polish cryptographer Marian Rejewski discovers how Enigma works and Alan Turing figured how to crack the key to designing the Bombe Machine as a decoder for Enigma encryption. The starting point of modern cryptography was Claude E. Shannon's article called "A mathematical theory of cryptography" [45]. Encryption became part of industry standards when IBM designed block cipher to protect the company's customers' data, using a deterministic algorithm operating on fixed-length bits blocks and based on a design principle of substitution–permutation network. It uses an unvarying transformation and symmetric key and was adapted by the US as a national standard called Data Encryption Standard (DES) [10]. Later DES replaced by

Advanced Encryption Standard (AES) [7] and a subset of block ciphers founded by Vincent Rijmen and Joan Daemen [11] which presented at open competition, it also accepted as US national standard.

Modern encryption methods are using a block cipher to disguise the meaning of the message and use extra variables as a key for creating the encoded message. There are two common approaches for the encryption methods, symmetric and asymmetric, When the symmetric ciphers requiring a single key and asymmetric ciphers use two different keys with a mathematical connection that one of them is public. For exchanging public keys without exposing, the Diffie–Hellman key exchange [13] provide satisfying technique to establishes a shared secret between two parties. Encryption algorithms often use prime numbers to create keys since it is computationally difficult to factor large prime numbers and reverse-engineer the encryption. The Rivest-Shamir-Adleman (RSA) [39] encryption algorithm is currently the most widely used public-key algorithm along with a new versions of DES, AES, and even hashing functions like HMAC. All encryption algorithms have to apply four steps: encodes the message's content, verifies the origin of a message, proves that the contents of the message have not been changed since it was sent and prevents senders from denying they sent this encrypted message.

2.3 Internet Security

The openness of the internet-network has always been known as vulnerable. With the spread of computer networks, the ability of users to send malicious messages in the network was a problem and tools like anti-virus software were developed to deal with known threats. Anti-virus software scanned all the binaries or searching strings that typically found in the malware and tested them against a database of trusted signatures to validate the reliability of the data. Years after, the number of new malware samples produced every day has grown enormously, which made it difficult for anti-virus software's handle network security. The requirement of improvements in computer-network security arrived in the shape of security-suites which includes Anti-virus software, Firewall, and spyware which makes signatures scanning for malware families, blocking the unwanted connection and detecting for information leaking or unusual use in the network.

While the development of computer-security has grown, it became harder to perform active cyber-attack and the importance of the passive one has increased. The way that passive cyber-attack acts make it very difficult to prevent or even identify. The hand-shaking procedure was implemented as standard in network-protocols to obtain authentication but we're not suggesting a proper solution for monitoring the network and sniffing packets by a hostile user to perform a passive cyber-attack. Therefore, encryption methods became a suitable solution for the threat of passive cyber-attacks. Encryption is a technique for sending messages secretly through a public communication medium, it involved modifying the message in a way that it can be in a certain way or after performing some calculations, called encoding, and decoding.

the abilities of encryption produced secure internet-network protocols such as HTTPS which use SSL/TLS encryption protocols for securely communicating over the internet-network. The secure socket layer (SSL) is an encryption protocol developed by Netscape and designed to provide privacy in internet-communication [16]. It authenticates between the server and the client and extends the standard reliable

transport protocol (TCP) for data transmission and reception. SSL protocol is an independent part inside the packet layer structure, so application protocols like HTTP can use the extended layer of the SSL without adjustments.

The protocol using an SSL certificate to verify the endpoints with trusted authority as a third-party server. It includes cipher-suite (encryption algorithms) and key parameters which negotiated while hand-shaking to configure session encryption. The process starts with the user applies for identity from the servers, the server sends the user a copy of its SSL certificate, the user checks if it trusted, infer the server, the server sends back acknowledge to start an SSL encrypted session. Once the handshaking is done, the data in the session will be encrypted and transmitted between the endpoint followed the agreed encryption configuration. SSL was never wildly used and did not adopt by a major number of internet services due to lacks, The latest version of the encryption protocol called TLS [12] is upgraded SSL protocol which uses hash-based message authentication code (HMAC) after each message encryption instead of normal MAC as SSL. TLS relies on Diffie-Hellman as public key encryption (asymmetric) for the key exchange process, with the assistant of known Certificate Authority (CA) server that's supplies a public key certificate for handshaking process. After authentication established secretly, data transfer will use private key encryption (symmetric). The combination between the two techniques been made to reduces latency and based on the assumption, which none of the sides need to expose the secret key, because it passed secretly while handshaking. As mentioned, TLS calculates secret key hash (HMAC) for each message to ensure integrity and finally creates Digital Signature for the specific message by encrypting the HMAC with the senders key. when the client and server can communicate secretly on their first interaction a lot of possible attack vector gone useless.

The awareness for internet security encourages organizations and private users to adapt improved security steps, like using a VPN and manually configuring the encryption methods (also called modify cipher-suite). A VPN is a private overlay network among distributed sites that operates by tunneling (routing) traffic over public communication on the internet-networks. A chosen VPN's private server is used instead of the original ISP so that when your data is transmitted to the internet, it comes from the VPN rather than your computer. This method hiding your IP address and using its encryption techniques for guaranteeing secure remote access to servers and services.

2.4 Machine Learning

Machine learning is a learning approach considered as the field of AI, based on a computer learns from experience and takes actions based on previous execution [26]. Attempts to computerize learning processes have a significant milestone when A. L. Samuel's applied learning procedures in the game of checkers [42], these procedures enabled the computer to improve from the status of a beginner to a tournament player by practicing. Over the years, the idea has generalized, and programs have a suitable learning procedure for solving problems automatically and more efficiently. This evolved to the AI field in the shape of machine learning, where the learning process, also called training, with a big data-set, constructing a model of machine language rules over a coordinate system or decision trees, and developed procedures for tasks like classification. The classification problem discovered useful in serious tasks [34], it involves learning from classified (labeled to categories) data samples

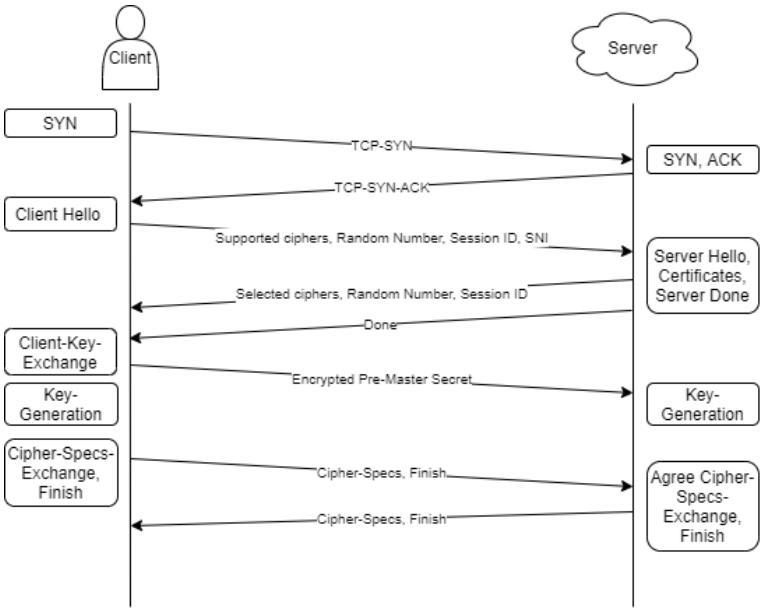

FIGURE 2.2: TLS/SSL Encrypted Protocol Handshake, After Server Done, Messages are Encrypted

and then distinguishing between data samples, which contains values for predefined features, and perform ascription for a known set of categories.

using machine learning algorithms for creating a classifier, that can be used for identifying a new sample of data efficiently, requires a couple of steps [22]. First is collecting a big data-set of appropriate samples, then choosing the most informative values in a sample for the classification task, and extract them as a data-set for learning, this called feature extraction. Third, pre-processing of the data to handle noise and maintain the mining quality while minimizing the sample size. Fourth will be feature transformation where an optimized set of features with the best classification performance is selected for the task. At last, the data became the input for the learning process (training), for classification algorithms like Support Vector Machines (SVM). SVM represent the samples in the coordinate system as points in space, with declared sub-spaces (areas) for each category, hyper-planes dividing the samples by a clear gap and maximizing the margin by creating the largest possible distance between the separating hyper-plane. Other than SVM, there are decision trees which are flowchart-like structure containing nodes with parameter condition, and connection referenced to the source node condition. Decision trees make computation from the root (first node) to the leaf (nodes at the bottom of the tree) and pave a path by the input values that are related to each node. Computing classification made by sorting nodes based on feature values, each node make a condition on a feature value of the sample and continue down the tree through the proper connection (branch), which represents a value that the node can assume. At last, it ends up at leaf node that represents a category in the classification.

2.5 Deep Learning

Deep learning [54] is an advanced field of AI, where the intention is to simulate the human brain processing when analyzing data and making decisions. Deep learning, using Artificial Neuron Network (ANN) implementation of artificial neurons connected, as weights modeled network for unsupervised learning and imitate the

operation of biological neural network, for dealing with unstructured or unlabeled data. The artificial neurons are processing elements with static and symbolic states for Informing response to external inputs. When constructing a weighted highly connected modeled network, usually with a lot of artificial neurons, it evaluated as ANN. Like machine learning algorithms, the ANN requires learning process as training over a big data-set, unlike machine learning, deep learning automatically discover the patterns needed for detection or classification tasks. Connected ANN can accomplish a learning process of machine learning, using multiple bounded size layers of ANN, where the deep part refers to hierarchy leveled features, created while learning. Each layer level learns to transform the input data to abstract and composite value.

The ANN was found to be an efficient in geometry field by Minsky [27]. When a linear perceptron could not perform as a universal classifier, and the neural network, by activating a non-polynomial number of functions with one hidden layer of unbounded width, found as proper classifier. Any deep learning systems have its credit assignment path (CAP) depth which is the chain of transformations from input to output, where each step represent a transition of information through a layer of neurons that intend to compute specific feature of the input data. The models can be structured as recurrent-neural-network where the data can pass through the layers couple of times before evaluating the result, and feed-forward-neural-network where the data passing layer once in the computing process. Feed-forward neural network is considered useful for supervised learning when, the back-propagation process offers a class of algorithms, that efficiently computes the gradients of the loss function for adjusting the ANN. Eventually it maps a couple of inputs variables into a value, representing the loss by combining the inputs and sets proper gradients, depends on the weights in the network. The back-propagation process includes applying a learning algorithm to train the network and adjusts weights. The ANN was firstly discussed by Mcculloch [25]. Examining the different behavior of neural networks for logical calculus, and back-propagation presented by Kelley [20]. When applying gradient theory for calculating optimal flight paths.

Convolutional neural network (CNN) [53], which also called shift-invariant ANN, are feed-forward-neural-network of regularized multi-layer perceptrons (MLP), that is a fully-connected network, where each neuron in one layer is connected to all neurons in the next layer and the computation operates by a convolution between the layers, instead of matrix multiplication. CNN discovered in the field of biological processes when the animal visual cortex resembled and the connectivity pattern between neurons examined [18]. Cortical neurons discovered to respond for a specific part of the input known as the receptive field, and all connected neuron together covers the full input data. The convolution operation [51] computes the integral of the product of the two functions, after one is reversed and shifted, and measure the influence of two mathematical functions on each other. It brings a solution to computations that require a large number of parameters and shrink the input size to much smaller output, which makes CNN a preferred ANN solution for fields like image processing when sample input of image consists of a lot of data.

The structure of CNN contains input and output layers as usual neurons networks and the hidden layers between those two can include convolution layers such as pooling layers, fully-connected layers, and normalization layers. The pooling layer [43] intends to reduce the dimensions of the data, it combines the outputs in neuron clusters at the previous layer and set then as a single neuron input for the next layer, where the max-pooling layer makes the combination of the neuron cluster followed the maximum value in the cluster.

The hidden layers of the CNN also including an activation layer and loss layer, where the activation layer maps the output values of previous later to specific input values for the next layer, and the loss layer for measuring the distance (loss) between the result of sample computation and the original result of the sample. Those layers depend on the objective functions they use where the ReLU activation function [30] is mostly used in CNN's structure to avoid negative values pass through the layers and enable better training of deep ANNs, where soft-max loss function [8] is known objective function of the loss layer. This network architecture supposes to deal with ANN over-fitting of training data, where the model is exactly adjusted by the trained data which, usually includes noise and can cause wrong computations. For image recognition, CNNs usually include one or more convolution layers, straightly followed by the ReLU activation layer and max-pooling layer before any convolution layer (except the first), where Soft-Max is the last layer for evaluating the output. It proved to be efficient in features learning as well as classifications task [23].

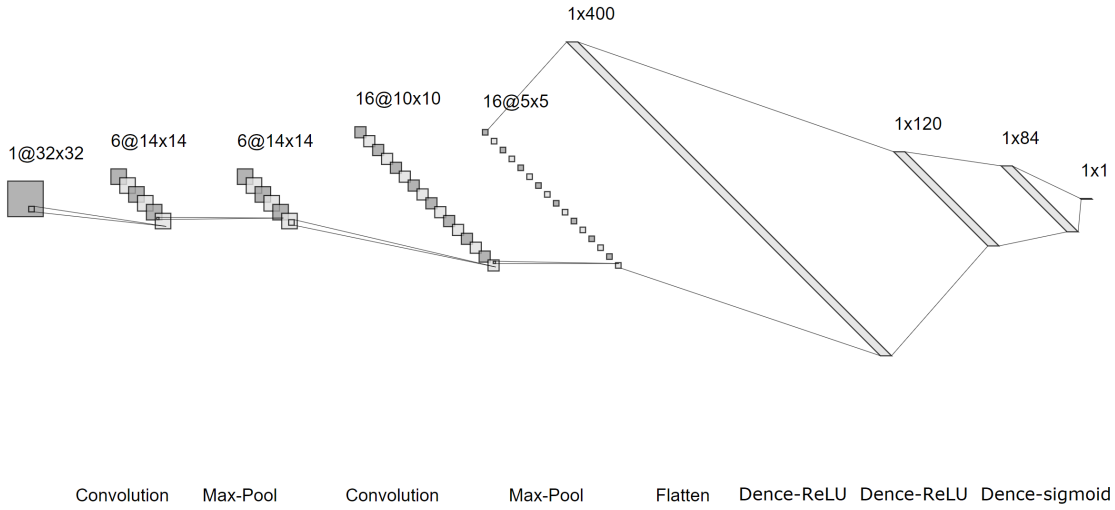

FIGURE 2.3: Convolution Neurons-Network Architecture

2.6 Robust Learning Model

Adversary Opponent, is considered as a malicious entity, whose aim is to prevent from targeted service to correctly compute information. For example, in cryptography, it can interrupt users of a crypto-system to achieve privacy, integrity, and availability of data. In machine learning, adversarial opponents are examined as an intervention in the learning process as well as in evaluation to cause failures in the model intention. Machine learning in an adversarial environment, includes an opponent that tries to cause machine learning to fail in many ways, manifested as attempts to break the assumptions learned while training. The real-world data has various weak stochastic properties, independence, and known distribution. Adversaries try to use known behaviors of legitimate data to make an impact in several manners, such as avoiding detection systems, causing miss-classification of legitimate input by targeting real-world behavior, discover exploits in the learned model and performing DoS on the machine-learning classifier. In many cases, the adversary opponent is considered as a direct user of the machine learning service, or third-party entity tracking machine learning model activity and be able to poison the learner's classifications.

An adversary can create wrong but similar data with feature properties act like normal data and use it as input for miss-classification.

Huang [17]. have studied adversarial cases against machine learning algorithms, suggested the ability to classify attacks against online machine learning algorithms, and explored vulnerabilities in machine learning algorithms. Introducing two models for adversary's capabilities, real-world attacks that spammers use against deployed anti-spammer like SpamBayes for labeling spam emails, and Anomalous Traffic Detection to achieve evasion by increasing the number of false negatives classifications. Biggio [6] defined adversary's knowledge levels include Perfect, Limited, and Zero-Knowledge. Perfect knowledge is defined as the adversary knowing the feature space, type of classifier, and the trained model. In the limited knowledge case, the adversary knows feature representation and the type of classifier without knowledge about the trained model. Lastly, zero-knowledge is when the adversary does not know any of the details of the machine learning system. Papernot [35]. Presented the limitations of deep learning in adversarial settings in image classification and used Jacobian Saliency Maps to identify the perturbed inputs. In the image classification field, Adversarial Machine Learning (AML) consist perturbation of an image by adding noise to cause miss-classification by the machine learning classifier, while still being correctly classified by the human eye. The focus of adversarial cases of machine learning, in image classification, is on the mathematical analysis of the boundaries and properties of a specific machine-learning algorithm, to identify miss-classification vulnerabilities.

A great number of possibilities for feature manipulations can affect an adversary opponent, when additional characteristic has been used to represent learning samples, the adversary opponent can find it more difficult to cause miss-classification. Using different combinations of features will make implementation times increased, therefore a combination of several features could apply robustness of the learning model against adversarial opponents. The robust model indicated when the classification output is consistently accurate, even when features values or assumptions are drastically changed, learning in the presence of outlines, and trying to supply privacy for data or user's information. Robustness testing already been viewed as fuzz testing which involves invalid or unexpected inputs, and fault injection as learning process interruption. In case of adversarial opponent, the robustness examined against the varied know adversarial situations including noise, statistics manipulations, padding techniques, etc.

Chapter 3

Related Works

The evolution of Artificial-Intelligence (AI) including big data and learning algorithms implies an affordable solution for analyzing encrypted traffic. Two distinct fields of AI have made their impact in classifying encrypted traffic mostly to categorized traffic types such as browsing, video-streaming, chatting, etc. and the application (AKA supplier) which can be Facebook, Google, YouTube, etc. The Adversarial Opponent (AO) case of VPN and Tor use has been considered when labeled data-sets with appropriate records were published. Additionally, generating manipulated data in several cases also represented. Two distinct approaches with proofed success utilized researches. First were supervised Machine-Learning (ML) that includes statistical decisions powered by classified big data-set over coordinate system or decision trees representation. Second, supervised Deep-Learning (DL) using a Neural-Network (NN) algorithm of different types and architecture.

3.1 Machine-Learning

The classification of BOA tuple has been reviewed earlier by Dvir [15], [29] when records of 20,032 different traffic sessions of encrypted internet-communication were collected [3], as.pcap files, for simulating the activity of hacker or service provider when tracking user activity in the network. Traditionally those records have been extracted into 53 numeric features to represent each session, via features extraction process [2] to produce a big data-set as input for learning algorithms. The features set were constructed from a combination of 6 groups of session features, SSL parameters, TCP parameters, size of packets, packets amount, packets arrival time, peaks of data transmitted, and statistics features about the mentioned groups. Two distinct learning models were used for the classification task, SVM in several configurations and Random Forest (RF), which is a learning algorithm that uses several Decision-Trees (DTs) at different sizes (means a different subset of features as nodes) to make a generalized decision. The training-testing process results in 96% of success using RF for classifying the BOA tuple. Furthermore, the fact that using only the first second of the session did not decrease accuracy dramatically and accomplished 94% success, again with RF.

A comparison of popular DPI tools for traffic classification for encrypted network-protocols has been made by Bujlow [9], and Nguyen [31] has surveyed different ML techniques. Bar [5] developed a statistical classifier that allows real-time classification of encrypted data that got an average accuracy rate of 83%, where Gil [14], used time-related features to characterize the network traffic using k-nearest neighbor (k-NN) and achieved accuracy levels above 80%. Alshammari [4] offered Skype encrypted traffic identification using Genetic Programming (GP) with 98% of success, Weina [32] presented the heuristic statistical testing (HST) approach that combines both statistics and ML with 91% of success. A hybrid approach that uses DPI for

extracting traffic features, powering ML algorithms for classification, results as the most efficient technique for the task. Those methods require extracting features like packets-size and packets-arrival-time to apply supervised learning.

Moore [28] provided features discriminators that describe the session in internet communication, and Shen [47], described a systematic approach of feature selection for the classification of encrypted traffic, also involved features about the session. The research marked the time features as less important for the classification, when the main features presented are backward-forward packets-size offset, statistics about packets-size, and a sequence of packets-length at the beginning of the session. The procedure divided into three parts, pre-processing of the features set, evaluate their importance facing the computation overhead, and combining features to construct an optimized set. In addition to the traditional model-based ranking as feature importance, the Chi-square test and Term frequency by Inverse document frequency (TF-IDF) are also examined in the evaluation part. They achieved optimized features set for encrypted traffic that are efficient over different data-sets. Nevertheless, the drawback of all these approaches is that the feature extraction and feature selection phases are essentially done with the assistance of an expert. Hence, these approaches are time-consuming and sensitive to mistakes.

3.2 Deep Learning

The DL field also raised efficiently and been used in several ways for classifying traffic, Ting [50] showed that the Artificial-NN model can perform better classification than Naive Bayes methods, where Wang [52] converted each packet payload to a normalized byte sequence, and used it as an input for Artificial-NN. Zeng [**zeng2019deep**] presented a deep-full-range framework for automatically learning raw traffic and classifying the service type using the DL method, when One-Dimensional Convolution-Neural-Network (1D-CNN) with Local Response Normalization (LRN) layers were used to achieve precision results of 98%. Follow the last methods, Lotfollahi [24] suggested a method combining features extraction and classification of application type using the UNB ISCX VPN-nonVPN data-set [**UNBDATA**] A "deep packet" framework developed with the use of 1D-CNN and auto-encoder for DL operation, results in 98% success for classifying the application when the CNN proved to be useful. The architecture of the network included two convolutional layers followed ReLU activation function and pooling layer. Then, a two-dimensional tensor is used to produce a one-dimensional vector ahead of a three-layered network of fully-connected-NN. It ends with a Soft-Max layer for preforming classification, while variant setups used, such as early stopping technique to avoid over-fitting in training (when the loss function remains unchanged after it found to be valid for several epochs), and dropout at rate 0.05 for setting some of the neurons to zero value, and Batch-Normalization.

Shapira [46] also deals with encrypted traffic classification, while using the well-known ability of CNN to recognize, they defined the "FlowPic", which presents a histogram graph of packet size along packet arrival time, where samples were regularized to 60 seconds of the session each. This research aimed to classify the service type and application name achieved 99.7% accuracy in the task. Using LenNet-5 architecture of the CNN model includes two convolution layers with ReLU functions and pooling layers right after, flatten layer, fully-connected network, and again ends

with SoftMax. The training process contained regular, VPN, and Tor traffic, performed categorical cross-entropy function of the SoftMax outputs to measure the results, with Adam optimizer [21] and dropout technique. The network became stable after 10-25 epochs, where any epoch took 5-10 minutes. Results were best performed after training each class separately against all others. Zhang [58] took an advanced version of CNN including capsule neural network (Caps-Net) which replaces the scalar feature output of CNN with vector output, and replaces the max-pooling with consistent routing, and disturbs some values in the capsule to reconstruct images. It marks that the reconstruction module can make the classification results easier to understand because of the gray-scale and achieved 98% accuracy for classifying service type of traffic. Rezaei [38] made a DL overview for classifying encrypted traffic and presented commonly used DL methods and their application in traffic classification tasks.

Salman [41] utilized two different ways for data representation as packet-based and session-based representation, compare the two representations for CNN-based traffic classification. The hierarchical classification framework of the traffic divides the classification task into categories such as classifying applications, user actions, and device types. Using 4 features as packet size, inter-arrival time, transport protocol, and direction for the first-N packets of the session traffic. Best scores were achieved by the session representation with 95.84% accuracy. Salman's previous work [**salman2018multi**] showed that CNN is preferred compared to other DL architectures such as Recurrent Neural Network (RNN), and Deep Neural Network (DNN). Furthermore, the advantage of using ConvNet architecture is marked versus other CNN architectures like GoogleNet, ResNet, AlexNet, etc. The CNN approach was found to be efficient in facing the manual feature extraction involved in ML and statistical methods.

3.3 Adversarial Opponent

In internet-network traffic classification, an AO can abuse the widely known implementation of networking protocol (Official Internet Protocol Standards i.e. RFCs), this leveraged the ability to cause miss-classification when an accurate implementation of legitimate network-traffic can be exported by the AO. For example, modify session information, like the number of cipher suites offered for communication, to appear as other legitimate traffic for ML classifier and result in a miss-classification. De Lucia [**de2019adversarial**] covered different levels of adversarial knowledge types, and with wide features set, suggested sufficient defense against AO as an internet-security scanning detection classifier with SVM.

Amir [**sadeghzadeh2020adversarial**] explored the robustness of 1D-CNN to classify the application against Adversarial Network Traffic (ANT). While defining ANT techniques use Universal Adversarial Perturbation (UAP) for generating samples in three different ways. AdvPad attack injects a UAP into the content of packets, AdvPay attack injects a UAP into the payload of a dummy packet, AdvBurst attack injects a specific number of dummy packets with crafted statistical features UAP into a selected burst of a flow. The precision score of 84%-98% presented for classifying applications like Skype, Hangouts, etc. and validate robustness with the three mentioned methods.

when using traffic characteristics like frequency of the TLS record sizes in each session, or the total number of bytes in the session, AO can make session traffic to be unclassified. Modifying the sequence of TLS record sizes being exchanged in

each direction, can match a pattern of other types of traffic. AO case considered when modifying the list of cipher-suites offered and extensions supported, this requires dealing with several features to accurately cause a classification error. Dvir [**dvir2016robust**] examined this AO case, and measure the robustness of the models, where the adversarial methods considered are user communicating through VPN or modifying the cipher-suite. They presented 83% success for dealing with VPN traffic and 94% for cipher-suite modification. Vincent [**taylor2016appscanner**] produced robust Identification of smartphone applications presented the app-scanner for fingerprinting smartphone apps and established identify apps with accuracy between 73% to 96%. Qin [**qin2015robust**] reduce the number of packets being processed using PSD of "Bi-flow" to capture the exchange behavior characteristics of the communication in the session and achieve VoIP application identification, AO case of Poisson sampling method, verified robustness of their model with identification accuracy above 97%.

TABLE 3.1: Relevant Works Table

Author & Ref	**Title**	**Description**	**Propose**	**Scores**
Zion [60]	Classification and enrichment of encrypted traffic Using Machine Learning algorithms	built network traffic features set for ML identifying the BOA tuple with 96% accuracy using RF. Adversarial VPN opponent with 83% for OS and the Browser.	OS, Browser, Application Classification with VPN	96% regular traffic and 83% with VPN
Dvir [**dvir2016robust**]	Robust Machine Learning for Encrypted Traffic Classification	classifying the BOA tuple considering adversarial opponent which use VPN and change chipper-suite. Achieved 96% accuracy normally, 81% with VPN and 91% modify cipher-suit. Used RF and SVM+MAP.	OS, Browser, Application Classification with VPN and modified cipher-suit	96% regular traffic, 83% with VPN and 91% modify cipher-suit
Qin [37]	Robust application identification methods for p2p and voip traffic classification in backbone networks	reduce the number of packets being processed identifying P2P and VoIP application. Created a model to aggregate traffic packets (Bi-flow) and PSD to capture flow dynamics. got 97% accuracy in application identification.	application classification	97%
Bar [5]	Realtime classification for encrypted traffic	statistical classifier for real time classification of encrypted data. Success with 99% using k-NN for specific applications.	application classification	99%
Zhang [59]	Robust identify zero-day application traffic	defined RTC using Bag of Flow (BoF), SVM, RF. About 90% true positive rate.	application classification	90%
Lotfollahi [24]	Deep packet: A novel approach for encrypted traffic classification using deep learning	using the UNB ISCX VPN-nonVPN dataset developed “deep packet” using 1D-CNN and autoencoder results with 98% success for classifying the application..	application classification	98%
Shapira [46]	FlowPic: Encrypted Internet Traffic Classification is as Easy as Image Recognition	encrypted Internet traffic classification of service type and application name recognize histogram graph pictures of traffic sample as “FlowPic”, presents packet size along packet arrival time achieved 99.7% accuracy in the task.	traffic type and application	99.7%

Shen [47],	Optimizing feature selection for efficient encrypted traffic classification: A systematic approach	created UTA system for systematic procces of featuring encrypted internet-network traffic	features	-
Gil. [14]	Characterization of encrypted and vpn traffic using time-related	time-related features to characterize the network traffic using k-nearest neighbor (k-NN) achieving accuracy levels above 80%.	traffic type through VPN	80%
Al-sham-mari [4]	Unveiling Skype encrypted tunnels using GP	Skype encrypted traffic identification using Genetic Programming (GP) with 98% of success	Skype identifica-tion	98%
Weina [32]	A heuristic statistical testing based approach for encrypted network traffic identification	heuristic statistical testing (HST) approach with 91% of success.	traffic type	91%
Wang [52]	The Applications of Deep Learning on Traffic Identification	Classifying the protocol converted each packet payload to a normalized byte sequence and used ANN. 95% precision with SSL	traffic type	95%
Zeng [57]	Deep-Full-Range : A Deep Learning Based Network Encrypted Traffic Classification and Intrusion Detection Framework	deep-full-range framework 1D-CNN with LRN achieve precision results of 98%.	traffic type	98%
Zhang [58]	Network traffic classification method based on improved capsule neural network	Caps-Net, version of CNN achieved 98% accuracy for classifying service type of traffic.	traffic type	98%
Vin-cent [49]	Appscanner: Automatic fingerprinting of smartphone apps from encrypted network traffic	Robust Identification in the shape of Smartphone App, appscanner for fingerprinting smartphone apps established identify apps with accuracy between 73% to 96%.	applica-tion classifica-tion	73%-96%

Amir [40]	Adversarial Network Traffic: Toward Evaluating the Robustness of Deep Learning Based Network Traffic Classification	robustness of 1DCNN to classify the application against Adversarial Network Traffic (ANT). AdvPad, AdvPay, AdvBurst attack precision score of 84%-98% presented for classifying applications	application classification	84%-98% in adversarial cases

Chapter 4

Methodology

4.1 Tools

Jupyter-Notebook [19], is Notebook Interface (IPython Notebook format) as a local website service, for executing Python code followed by documentation and advanced outputs view, present data science tool for making experiments and creating reports. For machine learning abilities, *Pycaret* [36] open source, low-code machine learning Python3 library provided appropriate functions for building, tuning, and comparing machine learning models, and allows us to analyze models performance, compare settings, and export relevant learning analytic plots. The open-source data analysis and manipulation tool, *Pandas* [33] python library, also contributed to the research with big data handling abilities. *Wireshark* [55] open-source packet analyzer was our samples recording tool that created .pcap files of encrypted internet-network traffic.

4.2 Data-set

BOA Labeled data-set [3] of encrypted internet traffic, collected in Ariel University Cyber-Lab includes about 20,000 classified records as internet-sessions, been used. Data Collection made by the Crawler code project [1] based on Selenium browser automation [44] developed to create proper automation scenarios for different interfaces and activities like YouTube streaming, Facebook posting, etc. Samples of internet-session, also called flow, is defined as an ensemble of packets with the same < Source IP, Destination IP, Source Port, Destination Port >, while focusing on the SSLport (443) to collect the encrypted data. The packets indicated by the mentioned parameters can also be recognized as backward and forward packets in the session.

The *features-extraction* process creates relevant features for the BOA classification of the traffic records, produce informed numeric CSV data-set of labeled session records, represented by proper *features-set* of traffic characteristics including packets size, packets inter-arrival time, the total amount of bytes and packets, statistics features based on the latest and packet header parameters such as TCP and SSL parameters. The features have been collected followed by related works in traffic classification, to feed supervised ML and construct a highly accurate, real-time, robust set for classification. Our features are based on accessible information on encrypted traffic records. The forward/backward session will also be used independently, along with sessions of constant prefix packets as samples, the lasts can decrease the feature extraction computation time while maintaining classification scores.

4.3 Experiments

First, producing relevant learning *scores* for *prediction* with several Machine-Learning (ML) algorithms for classifying the BOA, while our samples are featured session values of encrypted traffic. For any model, the *scores* parameters of Accuracy, Precision, Recall, and F1, indicates the performance of the models. *Scores* was evaluated in two different ML approaches and results also been validated with related works, manly in the field of classifying the application behind encrypted traffic. Next, for each learning approach, AO cases will be developed, to test the *robustness* of the models, including padding, statistics manipulations, training interruptions, and malicious samples injections. Our test consistently divides the data to 50% for training, 20% for models evaluation, and 30% for *validation*, also called testing with *unseen* data, which refers for *robustness*, to attend the performance of the models detecting data which is different from the data used for building and evaluating the model (training). It is important to mention that raw data representation suffers from traffic anonymization and the fact that many packet fields are data-dependent, we will examine the performance of several ML algorithms avoiding anonymized data. Furthermore, we will mention that session-based representation is sensitive to the number of packets used for classification.

$$Recall = \frac{TP}{TP + FN} \tag{4.1}$$

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{P} \tag{4.2}$$

$$Accuray = \frac{TP + TN}{TP + FP + TN + FN} = \frac{T}{P + N} \tag{4.3}$$

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4.4}$$

4.3.1 Data Features

Declaring features for the diverse types of encrypted traffic is important for the usability of our classification method. We want to *select* features that produce *universal* analyzation process for any data-set of encrypted traffic records. The features-set constructed while surveying related works. We categorized features subsets to characteristics groups, and understand the *coverage* of features that capable to expose despite encryption. The features which will be examined are *size-features*, for example, packets count, packets length, in addition to statistics features of the last carried as a minimum, maximum, mean, median, standard deviation, and variance. *time-features* measuring the inner-arrival-time of the packets in the session, taken only as statistics. The time-to-live (TTL) and number-of-keep-alive (KA) as *header-features*, along with *TCP-features* and *SSL-features*, also will utilized our method. Additionally, two interesting features as cumulative packets length and cumulative packets count in manners of forwarding and backward packets also considered.

To *optimize* the features-set, we adapted the techniques presented [47] such as *low-variance* check and *features-importance*. We will also calculate computation costs of the extraction, to minimize the overhead, while maintain high accuracy score. Additionally, for *features-importance* testing, we will adapt the *Chi-square test* for more realistic measurements.

#	**Feature Name**	**Group**	**Direction**
0	fSSL_session_id_len	SSL	BOTH
1	fSSL_num_extensions	SSL	BOTH
2	SYN_tcp_scale	TCP	BOTH
3	SYN_tcp_winsize	TCP	BOTH
4	size_histogram_1	SIZE	BOTH
5	size_histogram_2	SIZE	BOTH
6	size_histogram_3	SIZE	BOTH
7	size_histogram_4	SIZE	BOTH
8	size_histogram_5	SIZE	BOTH
9	size_histogram_6	SIZE	BOTH
10	size_histogram_7	SIZE	BOTH
11	size_histogram_8	SIZE	BOTH
12	size_histogram_9	SIZE	BOTH
13	size_histogram_10	SIZE	BOTH
14	fpeak_features_1	PEAK	FORWARD
15	fpeak_features_2	PEAK	FORWARD
16	fpeak_features_3	PEAK	FORWARD
17	fpeak_features_4	PEAK	FORWARD
18	fpeak_features_5	PEAK	FORWARD
19	fpeak_features_6	PEAK	FORWARD
20	fpeak_features_7	PEAK	FORWARD
21	fpeak_features_8	PEAK	FORWARD
22	fpeak_features_9	PEAK	FORWARD
23	bpeak_features_1	PEAK	BACKWARD
24	bpeak_features_2	PEAK	BACKWARD
25	bpeak_features_3	PEAK	BACKWARD
26	bpeak_features_4	PEAK	BACKWARD
27	bpeak_features_5	PEAK	BACKWARD
28	bpeak_features_6	PEAK	BACKWARD
29	bpeak_features_7	PEAK	BACKWARD
30	bpeak_features_8	PEAK	BACKWARD
31	bpeak_features_9	PEAK	BACKWARD
32	packet_count	COUNT	BOTH
33	min_packet_size	STAT, SIZE	BOTH
34	max_packet_size	STAT, SIZE	BOTH
35	mean_packet_size	STAT, SIZE	BOTH
36	sizevar	STAT, SIZE	BOTH
37	std_fiat	STAT, TIME	FORWARD
38	fpackets	COUNT	FORWARD
39	bpackets	COUNT	BACKWARD
40	fbytes	COUNT	FORWARD
41	bbytes	COUNT	BACKWARD
42	min_fiat	STAT, TIME	FORWARD
43	min_biat	STAT, TIME	BACKWARD
44	max_fiat	STAT, TIME	FORWARD
45	max_biat	STAT, TIME	BACKWARD
46	std_biat	STAT, TIME	BACKWARD
47	mean_fiat	STAT, TIME	FORWARD

Table 4.1 continued from previous page

#	Feature Name	Group	Direction
48	mean_biat	STAT, TIME	BACKWARD
49	min_fpkt	STAT, COUNT	FORWARD
50	min_bpkt	STAT, COUNT	BACKWARD
51	max_fpkt	STAT, COUNT	FORWARD
52	max_bpkt	STAT, COUNT	BACKWARD
53	std_fpkt	STAT, COUNT	FORWARD
54	std_bpkt	STAT, COUNT	BACKWARD
55	mean_fpkt	STAT, COUNT	FORWARD
56	mean_bpkt	STAT, COUNT	BACKWARD
57	num_keep_alive	COUNT	BOTH
59	fcipher_suites	SSL	BOTH
60	fSSLv	SSL	BOTH
61	mean_fttl	STAT, COUNT	BOTH
62	cum_plength	SIZE	BOTH
63	cum_pcount	COUNT	BOTH
SUMMARY			
GROUPS		DIRECTION	
13 SIZE		25 BOTH	
4 SSL		17 FORWARD	
3 TCP		17 BACKWARD	
16 COUNT			
20 STAT			

TABLE 4.1: Network Traffic Features Table

4.3.2 Machine-Learning Survey

A survey of non-deep Machine-Learning models has been made, including well-known state-of-art ML algorithms to *compare scores*, and presents improvements using modern models. Learning will be powered by different data-sets to evaluate scores. We also determine the appropriate data setup of the delegate learning model like normalized and transformed data to define the appropriate learning algorithms for the BOA classification task. Further examination, divided the tuple and study the classification of each part separately, to present the relevance of our method each of the tuple.

To feed Neural-Network (NN) and finds how the *Deep-Learning* (DL) approach is useful in BOA classification, a new minimal set of features, based on size-features will be structured. We will use *CNN* model of two convolution layers, two pooling layers, a fully-connected layer, and ends with the Soft-Max layer that is relevant for *image-recognition architecture* and classify the tuple from samples of *packets-size histogram* session samples.

4.3.3 Machine-Learning Robustness

When using a small set of features for session representation, for example using only TLS record sizes, the AO would only need to perturb the single feature to accomplish a miss-classification, therefor a *robust* set of features has been required. Using proper features-set will force the AO to manipulate wide features-set values, which is more difficult when the number of disparate features increases. Using a well-selected large features-set can be useful against AO with limited knowledge. The limited knowledge opponent aware of the features used by the classifier, but still need to determinate the right combination of features represents the session. Examining the *robustness* of ML models will include testing the effect of padding and statistics manipulations on our features-set. Evaluating classification scores for each of the relevant models, and measure the AO influence hierarchically.

Usually, the AO is external to the system and only able to create malicious data. On the other hand, an opponent with access to a released statistic model, or classifier, may test it to reveal information about the training data. Moreover, an opponent may attempt to manipulate the mechanism to reveal more information about the data and perform privacy violations.

AO operation usually categorized by computational bounds (in terms of time and storage resources), passively listening (eavesdropping) or actively corrupting data (Byzantine), And static behavior or adaptive. Our cases will include advanced AO operations when computable-active-adaptive opponents will be simulated. Measurements are taken when AO operates while training (*Causative*) covering the influence over the training data, when the AO perform only after the training process (*Exploratory*). We also note the influence when AO operation targeting specified label for miss-classification (*Specificity*) and when abusing classification intrusion point to flood false-negative samples (*Integrity*). Additionally, we will test the ability of AO to discover information about the data, or the classifier model (*Privacy*).

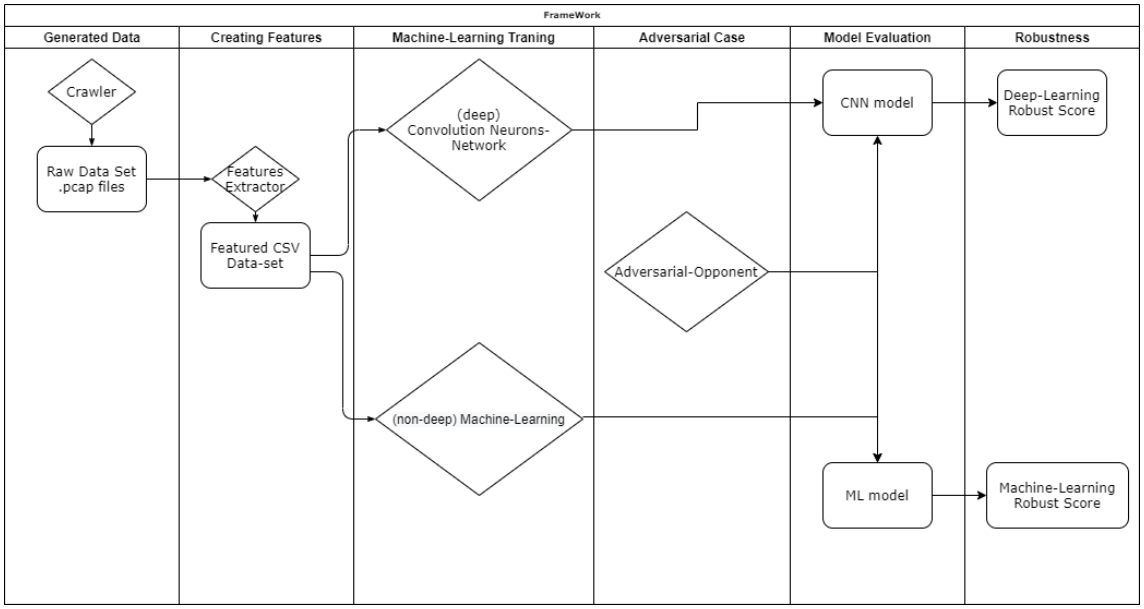

FIGURE 4.1: Research Workflow

Chapter 5

Primary Results and Conclusions

Results straightly concluded with related works, and represents *improvements* in classification scores. A survey of non-deep Machine-Learning (ML) models made [Fig. 5.1] to *compare-scores*. We also determine the appropriate data setup of the delegate learning model (linear models acts better with normalized and transformed data) to define the appropriate learning algorithms for the BOA classification task. Eventually, tree-based (EXBOOST, ET, RF) learning algorithms served as a well-performed classifier for our challenge.

Measuring the *robustness* of different features-set, with different learning models, powered by the selected features-set been made. Founds that it is necessary to keep the features that distinctly distinguish one class from another, while some of the features can be sensitive for the known AO cases. *Robustness* of the learning model can be reached, by using varied features-set, constructed from different session parameters. Features that are considered are a list of cipher-suites, packets-length, packets arrival-time, along with many other features.

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
Extreme Gradient Boosting	0.9804	0	0.8955	0.9799	0.9795	0.9771	0.9771	20.9745
Extra Trees Classifier	0.9795	0	0.9053	0.9795	0.9788	0.976	0.976	7.4156
CatBoost Classifier	0.9795	0	0.8895	0.9786	0.9784	0.976	0.976	392.6221
Random Forest Classifier	0.9727	0	0.8667	0.9724	0.9713	0.9681	0.9681	4.6086
Ada Boost Classifier	0.972	0	0.8794	0.9714	0.9708	0.9672	0.9672	63.8879
Light Gradient Boosting Machine	0.9472	0	0.8308	0.9715	0.9568	0.9393	0.9408	7.5602
Decision Tree Classifier	0.9401	0	0.8066	0.9557	0.9436	0.93	0.9306	4.3564
Quadratic Discriminant Analysis	0.8805	0	0.5718	0.8898	0.8735	0.8602	0.8616	1.8327
Gradient Boosting Classifier	0.8706	0	0.7586	0.9339	0.8733	0.8644	0.8744	274.0351
Linear Discriminant Analysis	0.8355	0	0.6335	0.8515	0.8278	0.8063	0.8078	3.9708
Ridge Classifier	0.7943	0	0.4293	0.7865	0.7503	0.7558	0.7595	0.831
K Neighbors Classifier	0.7737	0	0.5107	0.7643	0.7655	0.7348	0.7351	7.6802
Logistic Regression	0.4455	0	0.1173	0.3346	0.3497	0.2747	0.3034	8.8162
SVM - Linear Kernel	0.073	0	0.0502	0.1273	0.0715	-0.0129	-0.0141	1.5326
Naive Bayes	0.0564	0	0.0868	0.3369	0.0503	0.0334	0.0589	0.5724

TABLE 5.1: Machine Learning Algorithms Scores Table

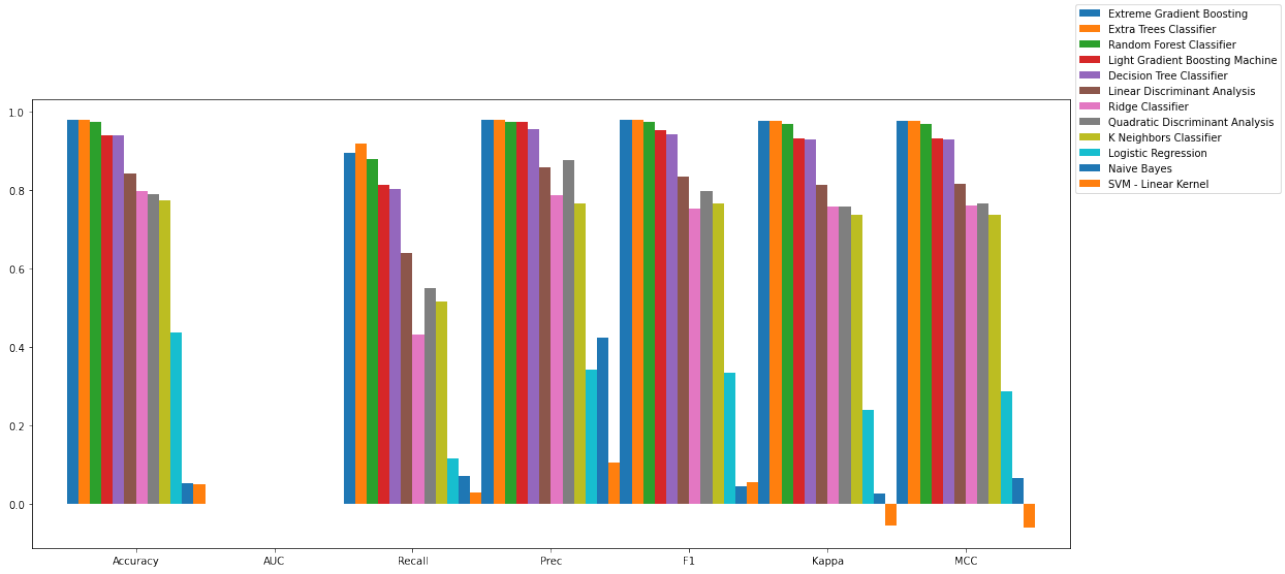

FIGURE 5.1: Machine-Learning Scores Graph

Bibliography

[1] Y. Zion R. Dubin A. Dvir J. Muehlstein and O. Pele. *crawler code.* 2017. URL: https://github.com/yzion/infomedia-crawler.

[2] Y. Zion R. Dubin A. Dvir J. Muehlstein and O. Pele. *feature extraction code.* 2017. URL: https://github.com/JonMuehlst/pcap-feature-extractor.

[3] Y. Zion R. Dubin A. Dvir J. Muehlstein and O. Pele. *research dataset*. 2017. URL: https://www.ariel.ac.il/wp/amitd/data-sets/.

[4] Riyad Alshammari and A Nur Zincir-Heywood. "Unveiling Skype encrypted tunnels using GP". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2010, pp. 1–8.

[5] Roni Bar-Yanai et al. "Realtime classification for encrypted traffic". In: *International Symposium on Experimental Algorithms*. Springer. 2010, pp. 373–385.

[6] Battista Biggio et al. "Evasion attacks against machine learning at test time". In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer. 2013, pp. 387–402.

[7] Eli Biham. "Advanced Encryption Standard". In: *International Workshop on Fast Software Encryption*. Springer. 1997, pp. 83–87.

[8] J Bridle. "Probabilistic interpretation of feedforward classification network outputs, with relationships to pattern recognition". In: *Neurocomputing: Algorithms, Architectures, and Applications. New York, NY: Springer-Verlag* (1989).

[9] Tomasz Bujlow, Valentín Carela-Español, and Pere Barlet-Ros. "Independent comparison of popular DPI tools for traffic classification". In: *Computer Networks* 76 (2015), pp. 75–89.

[10] Don Coppersmith. "The Data Encryption Standard (DES) and its strength against attacks". In: *IBM journal of research and development* 38.3 (1994), pp. 243–250.

[11] Joan Daemen and Vincent Rijmen. "The design of Rijndael AES". In: *Journal of Cryptology* 4.1 (1991), pp. 3–72.

[12] Tim Dierks and Christopher Allen. *Rfc2246: The TLS protocol version 1.0*. 1999.

[13] Whitfield Diffie and Martin Hellman. "New directions in cryptography". In: *IEEE transactions on Information Theory* 22.6 (1976), pp. 644–654.

[14] Gerard Draper-Gil et al. "Characterization of encrypted and vpn traffic using time-related". In: *Proceedings of the 2nd international conference on information systems security and privacy (ICISSP)*. 2016, pp. 407–414.

[15] Amit Dvir et al. "Robust Machine Learning for Encrypted Traffic Classification". In: *arXiv e-prints* (2016), arXiv–1603.

[16] Kipp Hickman and Taher Elgamal. "The SSL protocol". In: (1995).

[17] Ling Huang et al. "Adversarial machine learning". In: *Proceedings of the 4th ACM workshop on Security and artificial intelligence*. 2011, pp. 43–58.

[18] David H Hubel and Torsten N Wiesel. "Receptive fields of single neurones in the cat's striate cortex". In: *The Journal of physiology* 148.3 (1959), p. 574.

[19] Jupyter. *Jupyter interactive computing enviroment*. 2020. URL: https://jupyter.org/urldate={08/12/2020}.

[20] Henry J Kelley. "Gradient theory of optimal flight paths". In: *Ars Journal* 30.10 (1960), pp. 947–954.

[21] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[22] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. "Supervised machine learning: A review of classification techniques". In: *Emerging artificial intelligence applications in computer engineering* 160.1 (2007), pp. 3–24.

[23] Yann Le Cun et al. "Handwritten zip code recognition with multilayer networks". In: *[1990] Proceedings. 10th International Conference on Pattern Recognition*. Vol. 2. IEEE. 1990, pp. 35–40.

[24] Mohammad Lotfollahi et al. "Deep packet: A novel approach for encrypted traffic classification using deep learning". In: *Soft Computing* 24.3 (2020), pp. 1999–2012.

[25] Warren S McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.

[26] Donald Michie, David J Spiegelhalter, CC Taylor, et al. "Machine learning". In: *Neural and Statistical Classification* 13.1994 (1994), pp. 1–298.

[27] Marvin Minsky and Seymour Papert. "An introduction to computational geometry". In: *Cambridge tiass., HIT* (1969).

[28] Andrew Moore, Denis Zuev, and Michael Crogan. *Discriminators for use in flow-based classification*. Tech. rep. 2013.

[29] Jonathan Muehlstein et al. "Analyzing HTTPS encrypted traffic to identify user's operating system, browser and application". In: *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE. 2017, pp. 1–6.

[30] Vinod Nair and Geoffrey E Hinton. "Rectified linear units improve restricted boltzmann machines". In: *ICML*. 2010.

[31] Thuy TT Nguyen and Grenville Armitage. "A survey of techniques for internet traffic classification using machine learning". In: *IEEE communications surveys & tutorials* 10.4 (2008), pp. 56–76.

[32] Weina Niu et al. "A heuristic statistical testing based approach for encrypted network traffic identification". In: *IEEE Transactions on Vehicular Technology* 68.4 (2019), pp. 3843–3853.

[33] Pandas. *Pandas data analysis python library*. 2020. URL: https://pandas.pydata.org/.

[34] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up? Sentiment classification using machine learning techniques". In: *arXiv preprint cs/0205070* (2002).

[35] Nicolas Papernot et al. "The limitations of deep learning in adversarial settings". In: *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE. 2016, pp. 372–387.

[36] Pycaret. *Pycaret Machine learning python library*. 2020. URL: https://pycaret.org/.

[37] Tao Qin et al. "Robust application identification methods for p2p and voip traffic classification in backbone networks". In: *Knowledge-Based Systems* 82 (2015), pp. 152–162.

[38] Shahbaz Rezaei and Xin Liu. "Deep learning for encrypted traffic classification: An overview". In: *IEEE communications magazine* 57.5 (2019), pp. 76–81.

[39] Ronald L Rivest, Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems". In: *Communications of the ACM* 21.2 (1978), pp. 120–126.

[40] Amir Mahdi Sadeghzadeh, Rasool Jalili, and Saeed Shiravi. "Adversarial Network Traffic: Toward Evaluating the Robustness of Deep Learning Based Network Traffic Classification". In: *arXiv preprint arXiv:2003.01261* (2020).

[41] Ola Salman et al. "Data representation for CNN based internet traffic classification: a comparative study". In: *Multimedia Tools and Applications* (2020), pp. 1–27.

[42] Arthur L Samuel. "Some studies in machine learning using the game of checkers". In: *IBM Journal of research and development* 3.3 (1959), pp. 210–229.

[43] Dominik Scherer, Andreas Müller, and Sven Behnke. "Evaluation of pooling operations in convolutional architectures for object recognition". In: *International conference on artificial neural networks*. Springer. 2010, pp. 92–101.

[44] selenium. *Selenium browser automation*. 2020. URL: https://www.selenium.dev/.

[45] Claude E Shannon. "A mathematical theory of cryptography". In: *Mathematical Theory of Cryptography* (1945).

[46] Tal Shapira and Yuval Shavitt. "FlowPic: Encrypted Internet Traffic Classification is as Easy as Image Recognition". In: *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. 2019, pp. 680–687.

[47] Meng Shen et al. "Optimizing feature selection for efficient encrypted traffic classification: A systematic approach". In: *IEEE Network* 34.4 (2020), pp. 20–27.

[48] shobha1617. *Geeks-For-Geeks passive cyber attack definition*. 2017. URL: https://www.geeksforgeeks.org/active-and-passive-attacks-in-information-security/.

[49] Vincent F Taylor et al. "Appscanner: Automatic fingerprinting of smartphone apps from encrypted network traffic". In: *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2016, pp. 439–454.

[50] Hu Ting, Wang Yong, and Tao Xiaoling. "Network traffic classification based on kernel self-organizing maps". In: *2010 International Conference on Intelligent Computing and Integrated Systems*. IEEE. 2010, pp. 310–314.

[51] Vasiliĭ Vladimirov. "Generalized functions in mathematical physics". In: ().

[52] Zhanyi Wang. "The applications of deep learning on traffic identification". In: *BlackHat USA* 24.11 (2015), pp. 1–10.

[53] Wikipedia. *Wikipedia Convolution Neural Network*. 2020. URL: https://en.wikipedia.org/wiki/Convolutional_neural_network.

[54] Wikipedia. *Wikipedia Deep Learning*. 2020. URL: https://en.wikipedia.org/wiki/Deep_learning.

[55] Wireshark. *Wireshark packet analyzer*. 2020. URL: https://www.wireshark.org/.

[56] Fang Yu et al. "Fast and memory-efficient regular expression matching for deep packet inspection". In: *Proceedings of the 2006 ACM/IEEE symposium on Architecture for networking and communications systems*. 2006, pp. 93–102.

[57] Yi Zeng et al. "$Deep - Full - Range$: A deep learning based network encrypted traffic classification and intrusion detection framework". In: *IEEE Access* 7 (2019), pp. 45182–45190.

[58] Fan Zhang, Yong Wang, and Miao Ye. "Network traffic classification method based on improved capsule neural network". In: *2018 14th International Conference on Computational Intelligence and Security (CIS)*. IEEE. 2018, pp. 174–178.

[59] J. Zhang et al. "Robust Network Traffic Classification". In: *IEEE/ACM Transactions on Networking* 23.4 (2015), pp. 1257–1270.

[60] Yehonatan Zion. "Classification and enrichment of encrypted traffic Using Machine Learning algorithms". PhD thesis. Ariel University, 2018.