ENCVIDC: An Innovative Approach for Encoded Video Content Classification through Machine Learning

By

Faiqa Amjad

00000273801

Supervisor

Dr. Fawad Khan

Submitted to the Faculty of Department of Information Security Military College of Signals, National University of Sciences and Technology, Islamabad in partial fulfillment of the requirements for the degree of MS in Information Security

December 2020

ENCVIDC: An Innovative Approach for Encoded Video Content Classification through Machine Learning

By

Faiqa Amjad

00000273801

A thesis submitted in partial fulfillment of the requirements for the degree of MS Information Security

Supervisor

Dr. Fawad Khan

Department Of Information Security Military College of Signals

National University Of Sciences and Technology, Islamabad

December 2020

Certificate of Correctness and Approval

It is certified that work comprehended in this thesis "ENCVIDC: An Innovative Approach for Encoded Video Content Classification through Machine Learning, was carried out by Faiqa Amjad under the direction of Dr. Fawad Khan, for partial accomplishment of Degree of Masters of Information Security, is correct and approved.

Approved by

(Asst Prof Dr. Fawad Khan)
Thesis Supervisor
Military College of Signals (MCS)

(Asst Prof Dr. Shahzaib Tahir)
Thesis Co-Supervisor
Military College of Signals (MCS)

Declaration

I, *Faiqa Amjad* declare that this thesis titled "ENCVIDC: An Innovative Approach for Encoded Video Content Classification through Machine Learning" and the work presented in it are my own and has been generated by me as a result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a Master of Science degree at NUST
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at NUST or any other institution, this has been clearly stated
3. Where I have consulted the published work of others, this is always clearly attributed
4. 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
5. I have acknowledged all main sources of help
6. 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

_____

Faiqa Amjad,

00000273801

Copyright Notice

Dedication

"In the name of Allah, the Most Beneficent, the Most Merciful" Read in the name of your Lord who created. Created man from a clinging substance. Read, and your Lord is the Most Generous. Who taught by the pen.Taught man that which he knew not (Al-'Alaq 1:5)

With immense blessings of Allah, the Almighty, I dedicated this work to my exceptional parents, adored siblings, teachers and friends; whose tremendous support and cooperation led me to this wonderful triumph; and of course, to my beloved country Pakistan.

Abstract

With the ever increasing growth in the number of online video viewers on internet, the service providers are getting curious regarding the nature of content being revolved over the network without actively interacting with the network device in order to take action rapidly in support of their diverse security business objectives that may include detecting the pornographic videos, video cyber bullying, offensive as well as fake videos, most viewed content, user's internet profile and video content similarity etc. Due to the dispersion of encoded video streaming techniques, the network video traffic classification has turn out to be a challenging task to perform as devoid of the authentic decryption key, it is obstinate to comprehend the actual content viewed by the user. However, the current advances in machine learning has demonstrated the fact that encryption can also lead to certain information leak which yields promising results in determining the actual transmitted content. This research exploits the classical machine learning algorithms to propose a classifier truly for determining the encrypted video content sighted by users over diverse video sites for instance YouTube, Netflix and Dailymotion under the normal network conditions. This classifier foretells the content with an accurateness of more than 98% within a second and has the ability to execute all the network administration and sanctuary related business objectives.

Keywords: *Dynamic Adaptive streaming over HTTP, Encrypted network traffic classification, Machine Learning, Nearest Neighbors, QUIC, Random Forest, Video compression and encryption*

Acknowledgments

I am grateful to Allah Almighty for giving me strength to keep going on with this thesis, irrespective of many challenges and troubles. All praises for HIM and HIM alone. Next, I am grateful to all my family and especially to my parents. Without their consistent support and prayers, this thesis would not have been possible. I am very grateful to my Project Supervisor Asst Prof Dr. Fawad Khan who supervised the thesis / research in a very encouraging and helpful manner. I am also grateful to my co-Supervisor Asst Prof Dr. Shahzaib Tahir. As supervisor and co-supervisor, their support and supervisions have always been a valuable resource for me.I am also thankful to committee members who have always guided me with their profound and valuable support that has helped me in achieving my research aims.Finally, I would like to express my appreciation to all the people who have provided valuable support to my study and whose names I couldn't bring to memory.

Contents

List of Figures

List of Tables

List of Abbreviations

ML	Machine Learning
TCP/UDP	Transmission Control / User Datagram Protocol
TLS/SSL	Transport Layer Security/ Secure Sockets Layer
VPN	Virtual Private Network
HTTP	Hyper Text Transfer Protocol
AES	Advanced Encryption Standard
S/DPI	Shallow/Deep Packet Inspection
DCT	Discrete Cosine Transform
XOR	Exclusive OR
VEA	Video Encryption Algorithm
MVEA	Modified Video Encryption Algorithm
RVEA	Real-time Video Encryption Algorithm
AVC	Advanced Video Coding
HAS	HTTP Adaptive Streaming
DASH	Dynamic Adaptive Streaming over HTTP
MPEG	Moving Picture Experts Group
SVM	Support Vector Machine
AI	Artificial Intelligence

RL	Reinforcement Learning
DES	Data Encryption Standard
DL	Deep Learning
GPU	Graphics Processing Unit item[RF] Random Forest
DT	Decision Tree
ROC-AUC	Area under Receiver Operating Characteristic Curve
MAC	Media Access Control
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative

Chapter 1

Introduction

1.1 Problem Statement

Video has grown into a new and popular mean of communication among the Internet users. According to Sandvine report broadcasted in February 2020 on mobile internet traffic phenomena, the video traffic constitute 65% of the entire internet traffic. YouTube, Netflix and Daily motion, etc. are the most commonly used video streaming services all over the web. To ensure security, these streaming services have established techniques to encrypt their traffic, particularly videos streaming services in order to guarantee the privacy of user's online watched activities.

With the elevation in the sum of online video viewers on internet every day, the internet security operators are getting attentive to know about the type and nature of content being rolled through the network deprived of being directly engaged with the network device in order to accomplish their specified network security associated objectives. Moreover, digital media particularly video being the most popular media possesses capability to influence the minds of huge target audience. This, in turn, gives rise to numerous societal problems including cyberbullying, terrorist activities, child pornography, religious disputes, political affairs, etc.

At national level, cyberbullying is increasing with the equivalent rate as the internet nowadays. According to the statistics displayed in an International Crime Prevention Center (ICPC) report of the year 2020, nearly 73% of the individuals face cyberbullying through videos annually. Different organizations need to detect such content to take reasonable actions against them. In the recent years, videos have become a powerful

and robust information communication platform for terrorist as well as extremist groups. Therefore, military forces need to detect such content for taking appropriate actions at the right time. Moreover, government agencies require such a tool for enforcing certain laws and regulations for the use of internet by keeping in mind the mostly viewed content.

At the individual level, parents entail such a technique to monitor activities of their children over the network so that they can stop them from harmful or dangerous content which can spoil them.

The traditional users of the internet are of the view that their online viewed activities are hidden and segregated from the outside world as encryption or similar counter measures tools have been employed over the internet. However, in recent times, with the introduction of encrypted video traffic content classification techniques, the confidential information about the users can be extracted from the encrypted content as well. The existing techniques present in the literature pose some limitations, which affect the performance. These may include inapplicability to real-time network traffic, adherence to the compression nature of the traffic and no new data feedback to the already trained models etc.

Therefore, the availability of a robust encrypted video content classification technique is the desire need of today's society due to several factors leading towards the fulfillment of desired and required security related objectives of the internet network.

1.2 Goals

The chief goal of this study is to identify encrypted video content viewed by users on various streaming sites under the available network conditions without directly interacting with the network device with utmost accurateness and competence for fulfilling all the network security related business objectives of the target audience.

1.3 Thesis Objectives

Subsequent points present the foremost intents of this study:

- Analysis of the already famous techniques for encrypted video content identification in terms of their functionality, correctness and effectiveness.

- Crafting a novel procedure for discovering the encrypted video content viewed by users on various streaming sites under the available network conditions to accomplish all the network supervision and security related business objectives of the target audience.
- Interpretation of the results and propositions for the future work.

1.4 Thesis Contributions

Following major contributions are made by this research:

- Analysis of existing encrypted video classification techniques to identify their limitations.
- Exploring the feasibility of exploiting the encrypted nature of the network traffic through various features and techniques.
- Proposed a novel encrypted video classification technique "ENCVIDC" based on real world network conditions.
- Performance analysis of three traditional Machine Learning (ML) algorithms to determine the best model for the proposed classifier.
- Finally, we have presented a detailed contrast of ENCVIDC with the existing encrypted video content techniques to highlight the efficiency and accuracy of our new technique. This can lead towards the acceptance of our technique for the future assistance in network security context.

1.5 Thesis Organization

The thesis constitutes the following chapters:

- Chapter 1: The chapter encloses the preamble, aims, problem declaration, objectives, and contributions of the research work.
- Chapter 2: This chapter deals with the preliminaries such as fundamentals of network encryption, encrypted traffic analysis, video traffic encryption, encrypted

video traffic classification and ML etc.to give the intended users more understanding view for the new proposed scheme.

- Chapter 3: Literature examination as well as critical analysis of the existing encrypted video content classification techniques is enclosed in this chapter.
- Chapter 4: This chapter deals with comprehensive detail of the proposed technique from initial dataset collection to the final classifier evaluation.
- Chapter 5: This chapter encloses the outcomes of the proposed technique.
- Chapter 6: The thesis report is concluded here and directions for future work are being suggested.

Chapter 2

Background

This chapter includes relevant details required for the audience to comprehend the subsequent chapters. Here section 2.1 deliberates the network traffic encryption, section 2.2 discusses the encrypted network traffic classification. Section 2.3 contains details regarding encrypted video traffic classification. Section 2.4 debates on the chief ideas concerning ML and the ML techniques that are being employed.

2.1 Internet Traffic Encryption

If two parties wish to communicate over the internet without the participation of any third party, security needs to be assured. In this circumstance, security indicates that the communicating entities should be linked in such a way that the contact channel among them should be segregated from the outside untrusted world. If network traffic is communicated devoid of any sort of encoding process, the actual communicated content can be straightforwardly understood with any sniffer by anyone.

Different browsers now-a-days hire Transport Layer Protocols (TCP/UDP) with Security Layer Protocols (TLS is substituting SSL) [1] for data streaming services.TLS utilization has turn out to be much more prominent in recent times [2]. TLS covers three mechanisms for guaranteeing security i.e. privacy by symmetric encryption, integrity via hashing and authenticity via asymmetric encryption.Since HTTP has been lifted to HTTPS, all client-server transactions are encrypted over the internet. HTTPS appears for a website when it is protected by a SSL certificate issued by a trusted authority. If the website facilitates encoding, HTTPS automatically redirects the user's accessed

unsecured websites to encrypted websites.HTTPS adds an extra layer of sanctuary to user's data and web browsing activities.HTTPS do not hide the name of the user accessed webpage but hides all the sensitive information used by the webpage.

Quick UDP Internet Connections (QUIC) [48-49] is a common protocol use for data streaming purposes by certain browsers for instance Google, Chromium and Opera etc. and has the ability to support HTTPS traffic.QUIC is fundamentally related to the grouping of TCP, TLS and HTTP/2 implemented on UDP protocol. It is present on the top of UDP and suffers no constraints like TCP. The traffic is quite different and is addressed when browser-specific traffic analysis is being performed.

IPsec [51] stands for Internet Protocol Security having no port number. It represents a secure protocol suite and offers authentication, integrity and encryption packet-wise rather flow-wise in order to fulfill all the network's security requirements. It comprises of three protocols i-e IPsec Authentication header, encapsulating security payload and internet key exchange. The IPsec sender encodes the data at application layer afore directing them athwart the network. The receiver authenticates the data packets to ensure the integrity of data send by the sender. It functions in tunnel as well as transport modes. In tunnel approach, complete IP datagram is encrypted as well as authenticated where as in transport approach, only the payload of datagram is encrypted as well as authenticated. However, IPsec depends on a pre-shared key that can easily be steal by an attacker in a man-in-middle attack scenario. Table 2.1 shows the difference among IPsec and SSL in context of network security.

Virtual Private Networks (VPNs) are protected passageway for communiqué by using IP sec in most cases.VPNs may encode the data through 256-bitAES or any other widely known encryption technique and makes harder for anyone else to pinpoint or visualize user's data on a wireless internet network.

It is considered surprisingly straightforward to encrypt any sort of data, however not simplistic, by existing encryption techniques. Devoid of authentic decryption key, it is challenging to comprehend the content of any conveyed information.

Table 2.1: Contrast between IPsec and SSL

IPsec	**SSL**
Internet Protocol security	Secure Socket Layer
IPsec defines to be a set of protocols for providing network security	SSL is a protocol used with transport layer protocols for providing network security
It operates in the network layer of the OSI model	It operates between the transport and application layer
It secures a virtual private network	It secures the web transactions.
It employs symmetric key cryptography.	It employs asymmetric key cryptography.

2.2 Encrypted Network Traffic Taxonomy

The taxonomy of network traffic emerges to be the most significant task in todays' cyber security era. If network traffic is transmitted without any encryption technique, the actual communicated content can be straightforwardly comprehended with any sniffer by anyone compromising the privacy of the user over the internet to a greater extent.Encryption is therefore becoming pervasive at present's Internet services leading towards secure communications between the server and the client. Devoid of decryption key, it is challenging to get the content of any conveyed information.

Many researchers in [3-5] have put forward their machine learning techniques for encoded network traffic cataloging by means of using the information leak caused by encrypted network traffic. Deep packet inspection [52] habitually fails in this case as everything is encoded. Exploration in this field assists in flowing traffic management as well as in enhancing security all over the network.

2.3 Encrypted Video Traffic Investigation

Video has become a new and popular means of communiqué among Internet users. Video streaming makes a great portion of internet traffic and it is rapidly growing day by day. YouTube, Netflix and Daily motion etc. are most frequently used video streaming services all over the web. They have established procedures to encrypt their

streaming services in order to guarantee the privacy of user's online watched activities. Consequently, network management related information retrieval by traditional deep packet inspection becomes an impossible task to perform. However, identifiable patterns can be detached from the encrypted traffic leading towards the exploitation of the user's privacy.

2.3.1 Video Traffic Encoders

Video cryptographic techniques are defined on the basis of specific method of encrypting the video content information. The digital videos are generally observed in compressed state as they are very heavy and lengthy in size. Video encoding typically necessitates the encryption system to be of little computational complexity and format biddable. Consequently, due to greater computational expenditure and power, it's not realistic to encipher video streams with contemporary cryptosystems.

Most frequently use video encryption techniques consist of Video Encryption Algorithm (VEA) [9] [16], its tailored editions i-e Modified Video Encryption Algorithm (MVEA) as well as Real-time Video Encryption Algorithm (DES IDES for encryption) and H.264/AVC.

The initial three mentioned procedures are grounded on the symmetric key encoding of definite factors principally encompassing the coefficients of DCT function. Original video data cannot be retrieved without the VEA secret key. Therefore the sanctuary solitary depends on the length of the secret key. The system can easily be broken down if this key size is small. A large-sized key is often infeasible. Moreover, these hire simple XOR (+) operations due to which the corresponding plain and cipher text pairs can be used to find the secret key easily by anyone.

H.264/AVC is a widely adopted video data hiding standard to avoid the leakage of information regarding the actual content of the streamed video. It employs the block-wise encryption of bit wise XOR (+) operation (like standard stream ciphers) and water marking (code term replacing) while maintaining the actual video file size and backward compatibility severely.

The video cryptographic techniques can be classified into four classes, which are presented in Table 2.2 along with the important features. Choosing the most suitable

Table 2.2: Video Encryption Techniques

Encryption Technique	**Fully-layered**	**Permutation-based**	**Selective**	**Perceptual[9]**
Mode	Every byte	Elected bytes	DCT coefficients	High-order bi-planes
Security	Safest	safe	Highly unsafe	unsafe
Complexity	High	Low	Low	Customized
Speed	Very fast	fast	Very fast	Customized speed
Algorithms	DES,RSA	Zigzag	H.264/AVC	Lain,Wang,Sung and Wang[7-8]

video encryption scheme depends upon the size of the video, streaming time, security requirements and the required network as well as computational resources.

Different video streaming sites such as YouTube now-a-days practice HTTP adaptive (HAS) as well as dynamic video streaming services over different browsers. HAS includes the ability to tailor a multimedia stream output signal to the actual network constraints of the clients. DASH [10] is dynamic adaptive streaming over HTTP which takes data over HTTP and employs TLS security protocol for hiding video content. DASH is not a protocol but uses TCP as transport layer protocol to direct the data. DASH requires numerous multimedia information bitrates chunks to be enabled upon the requesting server. DASH works by encoding a couple of seconds' long small chunks of videos several times under different bit rates and resolutions. These chunks are then placed on a server and client can access them through a video player by HTTP GET requests according to the network conditions. Due to this reason, the quality of the chunks may differ. The client's video player organizes the deliverance of the data in a fix sequence. MPEG-DASH [11] employs Media Presentation Description (MPD) to have all the chunks aligned with each other in a sequence.The addition of HTTPS adds overhead to the communication process between the server and the client. Though, this overhead is too low and cannot really affect the analysis process.

2.3.2 Video Encryption Techniques Challenges

This subsection highlights certain major challenges in video encryption techniques since these are vulnerable to several security threats and compatibility issues.

Some of the issues are summarized below:

- The combination of DASH with Variable Bit Rate (VBR) aid in the construction of video segments that are unique for every video. These segments cause an information leak known as a video fingerprint. An attacker can easily construct a video fingerprints database to figure out the type of content being watched over the victim's network. Gu et al. in [57] proposed an efficient way to extract some useful information from DASH through eavesdropping passively.

- The traditional streaming services consider a video as a single same quality heavy streaming file and is troublesome in the case of less bandwidth network.

- Video compression and encoding strategies can significantly affect the overall video analysis process, for instance, the fast scenes in a video are normally encrypted with complex and higher bitrates as compared to all other scenes.

- Usually, video traffic is classified at the application layer. If the prominent features of this encrypted traffic are correlated through any AI technique, evidence about the actual content of the video's stream can be leaked.

- Previous research [13-14] implies that at the network layer or physical layer, an adversary can fully comprehend the encrypted video stream without having a direct access to the victim's network device. Almost all video streams can be distinctively classified by their statistical features as these are unique for each video stream.

These issues can be employed by researchers in Machine Learning (ML) as input to conduct encrypted video traffic analysis in many aspects for instance encrypted video flow detection [13-15] from normal encrypted traffic, video resolution and bitrates prediction [16] to enhance the online watching experience of the viewers, quality of experience [17-18] for the betterment of the streaming services and YouTube video source identification [19] etc.

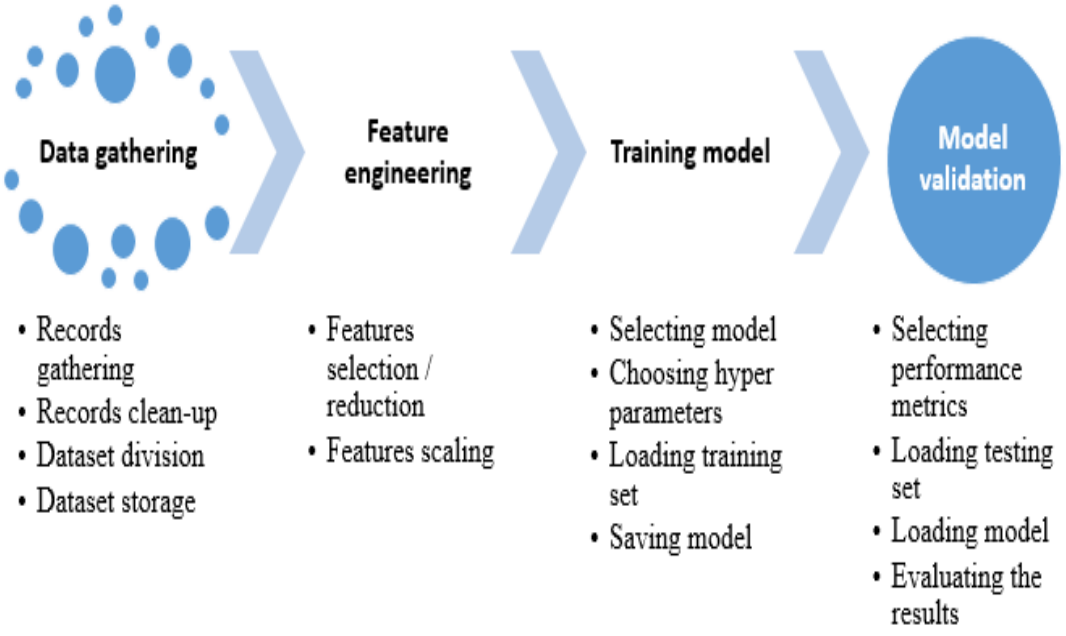


Figure 2.1: Machine Learning Steps

2.4 Overview of Machine Learning

Systems are not smart by default. ML [20-21] presents the division of Artificial Intelligence which offers systems the proficiency via knowledge through practicing robotically whilst being configured explicitly [22].We expect systems to be configured in such a way that they can train from the input available.

According to Herbert Simon, the performance of a system is enriched by learning from experiences. These experiences falls into two categories i-e direct and indirect experiences based on the input versus output relation in a system. The direct experience may include the one-to-one relation among the input and outputs of the machine whereas in an indirect experience, no such relation is defined.

ML is founded on some examples as inputs where each example comprises of a features set. A learning algorithm practices these features to deduce any sort of knowledge from them. Once this algorithm gets itself skilled, it can then forecast any analogous or interconnected data in the future. Figure 2.1 illustrates the general steps involved in ML.

ML has indeed been categorized in to several subspecialties on the basis of various forms of learning chores:

- Supervised Learning
- Unsupervised Learning
- Semi supervised Learning
- Reinforcement Learning

2.4.1 Supervised Learning

In Supervised ML technique [23], the machine learns from data that is already assigned a target class i-e tries to find relationship in the form of a function f between input and output of the machine. Labeled training data set T consists of an input x and output y as vector spaces for features and data classes. Here y is basically the description of its corresponding input component of x.The amalgamation of the entities in x with y vector spaces forms input examples of T.These input examples are supplied to learning algorithm for training. Once algorithm is trained, it can predict the target of any input data effectively in future. In simpler words, algorithm is trained in such a way that a particular instance from x will always Most common steps for classification through this library are as follows:map to the particular tag value of y. This linkage of x to y vector spaces assemblies like a model and can be defined as:

$$y = f(x) \tag{2.4.1}$$

Training ceases once the algorithm reaches a reasonable output performance level. Two methods are included beneath the umbrella of supervised ML based on their discrete and incessant nature of the outcome values:
Classification:It means the assemblage of the classifier's yield with a target value from y i-e plots x against the values of y.Given (x1,y1),. . . ,(xn,yn),a function f needs to be trained in order to predict y from x (y here is a categorical value)

The chief gain of using such a scheme is its extraordinary accurateness but at the expense of huge amount of training data against each class, high computational effort and time.Classification algorithms may include SVM, Random Decision forest and Naïve Bayes etc.

Table 2.3: Algorithm for Decision Tree

Algorithm
Data samples are firstly allocated to Root node
Let RN be the Root node
Using RN do:
discover f (feature value)and the associated t (threshold value) to split RN's examples in to Splitleft and Splitright
allocate (f,t) pair to RN
If Splitleft and Splitright are undersized :
Affix child nodes Childleft and Childright to RN and label them with the labels of Splitleft and Splitright
Else:
Affix child nodes RNleft and RNright to RN and label them with the labels of Splitleft and Splitright
Replicate the process by taking RN = RNleft and RN = RNright

2.4.2 Supervised Ensemble Random Decision Forest

RF [24] is a supervised ML technique that consists of a mixture of independent DTs. A single DT represents the weak classifier and has got vote to give in the final classification of input test data. Here RF will be treated as the strong classifier. The guessing inaccuracies are abridged with a combination of results from DTs. A divide and conquer strategy is often used by DTs to categorize data [25].Table 2.3 shows the construction of a DT for RF.

Characteristically, x features are employed in every single fragmentation of node when total x features are being employed in a taxonomy problem. Boosting and bagging are most commonly used terms when DTs are discussed.RF is basically a modified version of bagging.

Bagging [26] is defined to be an algorithm that mingles the yield from every decision tree in order to guarantee a more precise ultimate conclusion of the complete model. The vote from each DT is of equal weight and none can be ignored.DTs are adaptive to the particular data they are being educated on. The resultant DT can be quite diverse if

Table 2.4: Algorithm for Bagging Process

Algorithm
Let number of weak classifiers = 1. . . d:
pick examples random sampled subset n from training dataset D
Train weak classifiers Ce
Model combines results from C1 ..Cd

Table 2.5: Algorithm for Boosting Process

Algorithm
pick examples subset n from training dataset D to obtain D1
Train weak classifier C1 with D1
pick examples subset n2 from training dataset D containing almost half false outcomes to obtain D2
Train weak classifier C2 with D2
Take those examples of D whom both weak learners diverge on (D3)
Train weak classifier C3 with D3
Model is the outcome of weak classifiers

the learning input is altered in some way and the predictions are quite different in turn. Bagging involves parallel weak classifiers training by random sampled training set and simple average of the results from each DT when subjected to testing data. The deviation of the final model is reported to be diminished by bagging without any intensification of partiality. Bagging can result in highly interconnected DTs. The general algorithm for bagging is shown in Table 2.4.

On the other hand, boosting [27] which is also known as "Cleverest averaging of DTs" is defined to be an algorithm that builds new classifiers on the basis of previous classifier's false classification outcomes in a serial manner. Table 2.5 shows the typical steps followed by a boosting algorithm.

RF overawes over fitting difficulty via merging outcomes of diverse DTs.These are highly lithe and accurate in nature.RF efforts fine even if data values are absent or not scaled.RF is able to grip immense amount of data and the corresponding feature values

Table 2.6: Algorithm for kNN

Algorithm
An integer “k” is specified along with a testing sample
Selection of k samples that are closet to that testing sample
Calculation of the common classification from the samples
The result of the classification is the new predicated class for the test sample

effectively.RF is habitually sturdy enough to grip faults. However, RF's intricacy of construction leads towards high computational power, effort and time. The demerits described can easily be overcome using a more powerful system for construction.

2.4.3 k-Nearest Neighbors

kNN [47] is an easy to contrivance yet significant lazy ML algorithm that performs the task of classification and regression on different data samples using a distance and feature similarity parameter. The distance parameters may include Euclidean, Manhattan, Hamming and Minkowski distance. kNN works on the assumption that similar data samples exist in the same area.Table 2.6 encloses the steps for building up a k-Nearest Neighbors classifier.

In the classification process, a sample gets categorized grounded on the majority vote from its defined neighbor. In the regression process, a sample is given the average value of its k-neighbors i-e it is the approximation of incessant values. Small number of neighbors can lead towards noise in the classification problem.

The classifier is highly subtle over the structure of the data. The number of nearest neighbors depends on the data. Higher the number of neighbors can reduce error in classification problem but can lead towards high computational effort, cost and time. The outcome of the classifier usually become unhinged when number of neighbors are decreased to 1.The classifier can easily learn new data. It is not suitable for a high dimension of data. It is highly sensitive to missing and error values.

2.4.4 Support Vector Machine

SVM is a supervised ML classifier that performs classification as well as regression of single as well as multi class problem by employing a function called kernel. The basic function of a kernel is to enhance the dimensionality of the input data spaces and allowing them to become linearly separable. On the basis of the kernel function, SVM is considered linear or non-linear in nature.

The technique usually draws a line to separate samples to different classes. This line is basically described in context as "decision boundary". The samples on either sides of this boundary represent different classes. If the number of samples are two then the margin become a line and if the number of samples are greater than 2 (say n) then the margin becomes an n-dimensional plane. There can be a lot of lines drawn but the actual selected margin will be the one whom represents the maximum distance between the classes. The dimensions of this line truly depends on the feature values being supplied. If the data points are not near the decision boundary then there will be no misclassifications.

SVM do work well with clearly separable data but not for a bigger dataset as they require a huge training time. The computational effort, time and storage increases when the number of training support vectors increases. It is highly recommended to scale the data before SVM process starts as it is not scale invariant.

Regression:It means to predict output value using T records i-e plots x to domain of real values.Regression algorithms may include linear, logistic and generalized regression.Given (x1,y1),. . . ,(xn,yn),a function f needs to be trained in order to predict y from x (y here is a real value)

2.4.5 Unsupervised Learning

Unsupervised learning [53] implies the practice which trains a system with unlabeled data records i-e is forced to learn from itself. Here the training set T contains x but no y vector spaces.Given untagged x1,x2,. . . ..,xn ,output y will be observed from the patterns of x.

Two methods are included beneath the umbrella of un-supervised ML.

Clustering:It encompasses uniting untagged samples as analogous clusters. The clus-

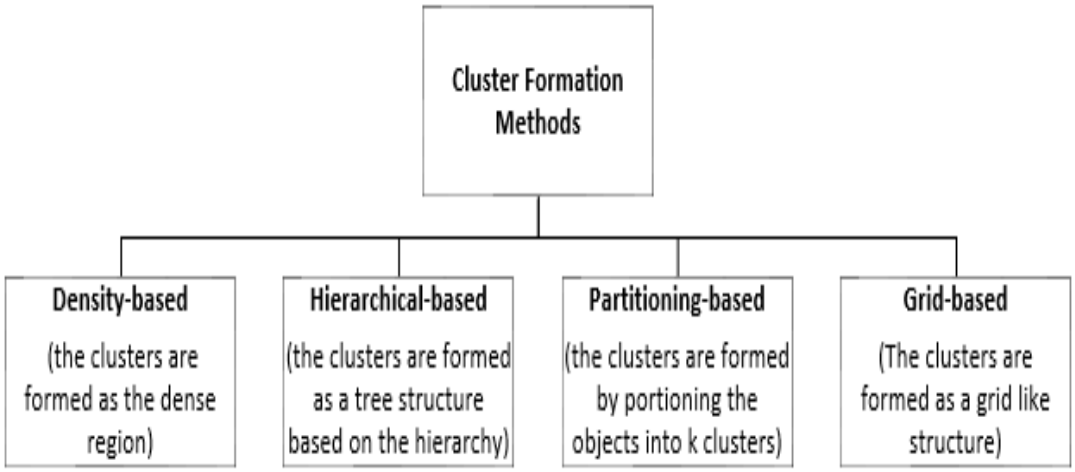


Figure 2.2: Types of Clustering

ters are assemblage of related samples. The principal objective is to catch correspondences from data points of a cluster.Figure 2.2 shows the different clustering techniques such as partitional, hierarchical, agglomerative and divisive clustering.

Abnormality Discovery:It is the technique to identify any sample that differs from majority of data samples. For example data irregularities for fraud detection in an online system.

2.4.6 Semi Supervised Learning

It falls among Supervised and unsupervised learning algorithms.T consists of labeled as well as unlabeled samples. Originally, semi-supervised learning [28] is driven by its significant significance for quicker, better and cheaper training. Majority of the circumstances involves sparse tagged data samples and profuse untagged data samples. It is predominantly valuable in case when the mining of features is problematic. It takes a partial labeled samples dataset for learning itself. As a result, this partial trained classifier is assigned the task to predict untagged data. The outcome in this case is referred as pseudo output. Practical applications include internet content classification etc.Let samples with X = x1. . . ,xn as well as corresponding tags Y = y1. . . ,yn and u

unlabeled examples Xu = x1+u. . . ,xn+u are taken together. Semi supervised learning aims to get possible outcomes for Xu.A mapping needs to induce between Xu and Y. Two methods are included beneath semi-supervised ML.

Transductive Learning:It is a learning technique that does not simplify to unobserved data and foresees specific samples given specific samples from a problem area. It seems to be a quick process.

Inductive Learning:It is a learning technique that simplifies to unobserved data i-e use tagged data to train a supervised learning algorithm and then use the predicted labels for the untagged data. It seems to be a slow process.

2.4.7 Reinforcement Learning

Reinforcement learning (RL) [54] involves intriguing appropriate acts in order to maximize recompense when subjected to specific circumstances serially. RL basically involves learning from indirect feedback after some data samples. In AI, wherever human influence is predominant, RL serves and performs smoother for example chess game. No supervisor is required here.Two basic concepts are included in Reinforcement learning:

Positive Reinforcement Learning:It is interpreted as once an episode arises owing to a specific action, the action's concentration and incidence boosts affirmatively.

Negative Reinforcement Learning:It is interpreted as the behavior's escalation due to a bunged or shunned pessimistic state.

Figure 2.3 shows the key components of RL.

Here agent is the one who performs actions on environment for gaining a reward. "Environment" here refers to the different events faced by the agent.RL assists in finding the circumstances where an action has to be performed. A sequence of actions, state and corresponding reward will be given to draft a function. This function is basically a mapping from the states to the actions. Table 2.7 highlights RL's algorithm.

2.4.8 Scikit-Learn Library

Scikit [46][55-56] is a well-known free python language based ML library which employs Numpy and Pandas for performing different operations. It basically contains various algorithms for classification, regression and clustering etc. via a python interface.

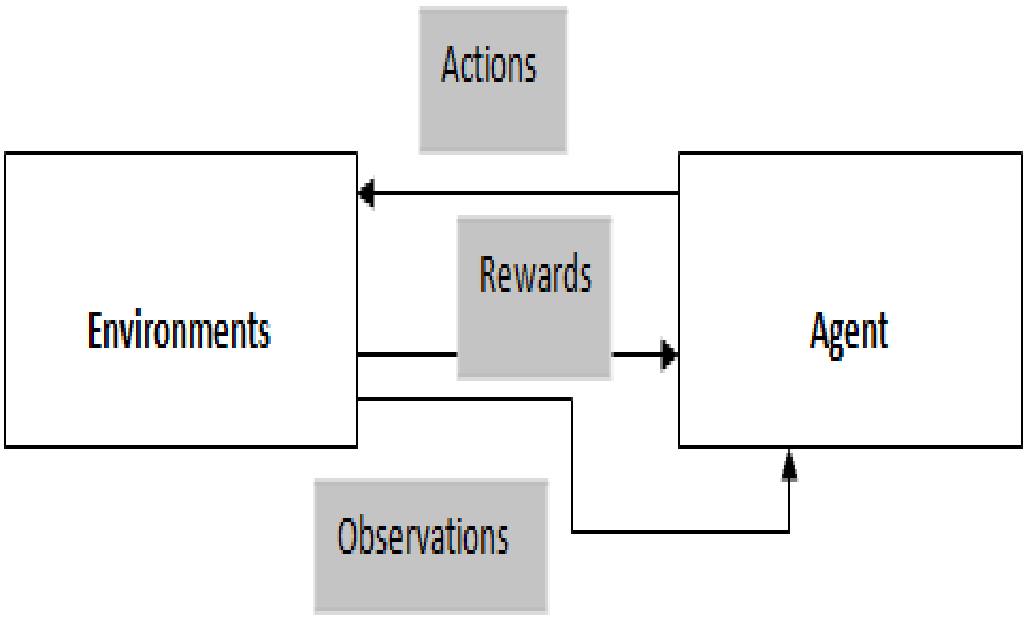


Figure 2.3: Reinforcement Learning

Table 2.7: Reinforcement Learning Algorithm

Algorithm
Let a agent, environment, action, reward and state be donated by A,E,a,R,S respectively
A interacts with E at distinct time intervals t where t = 0,1,2,. . . . . n
A monitors state S at t
A then generates a at t
A acquires resultant R
Repeat the process for next states

Most common steps for classification through this library are as follows:

- Choosing an appropriate model from the library based on one's requirement and nature of the classification process.
- Choosing model optimized hyper parameters to well-suit the training data. These hyper parameters can easily be figured out using three search methods:
  - Grid search investigates particular set of parameters of ML algorithm comprehensively.
  - Random search investigates a single parameter value autonomously via likelihood division.
  - Bayesian search investigates through the construction of a probabilistic ML model on the basis of factors like accuracy and mean error etc.
- Arrange, scale and reduce data in to features vector as well as target class vector. Sometime different type of data values are encountered by a ML algorithm. It is difficult for the algorithm to practice such diverse. One possible solution is scaling the data first and then processed further. The most common used data scalars include Standard Scalar and Robust Scalar etc. Some of these scalars are sensitive to errors. The basic rationale behind the use of this scaling is to translate the values in a standard form and same unit.
- Fitting the selected model by the data by using the functions from the library.
- Applying and validating the new data using the functions from the library.

The library provides the "Pipelines structure" to ensure parallel and fast processing. However, these pipelines can show many inadequacies based on for instance data quality, consistency and applicability etc. Accessibility and uncomplicatedness are the two chief pros of this library.

Chapter 3

Literature Review

This chapter comprises the details regarding the prevailing encrypted video content traffic classification practices and divergence among them in order to construct a more reliable scheme.

3.1 Existing Encrypted Video Traffic Content Classification Techniques

AI is a renowned scheme employed by a variety of researchers for encrypted video content classification. We have studied associated work that has tackled the encrypted video content classification by encrypted video streams. The video content can be classified into general categories i-e sports, play, pornography [29] [30], spam, fake, offensive [31] and restricted etc. as well as into particular categories for instance names of seasons, shows and songs etc. The techniques taking the particular content titles as input usually fail due to variable naming scheme over the internet.

Dubin et al. [32] established a classifier for YouTube's video general content classification over DASH [10] employing SVM [33] and kNN for the Chrome browser. Titles are the video's general content name like news, sports and education etc. The technique exploits the VBR phenomena of video stream's DASH. The traffic streams are fed into a filter that splits the new and registered traffic centered on the IP header information. Then YouTube streams are specifically removed from other encrypted traffic with the help of the string "googlevideos.com" in the client-server session initiation message in the first two steps and then processed further. The retransmitted and lost packets are

detached as they can cause classification problems. The audio streams and video streams are separated in the preprocessing phase on the basis of entropy values. Both are found in the same flow tuple due to which it is tricky to discriminate between them and are not in a sequential order. The preprocessed data is then input to the feature extraction process. The video streams are then combine into a peak. All the peaks of the streams are being taken at a time and features are extracted from them i-e the number of bits per peak (BPP).BPP are basically the ON/OFF patterns of videos. The feature vector must contain 40 peak features else it is pad with zeros to get the maximum length. These vectors are prepared consistently by applying Gaussian zero mean and unit variance. The NN algorithm calculates the likeness score among a testing title stream and training streams. Considering the feature sets, the matching score is basically the intersection of the feature sets having a maximum threshold value of 2.If the value of the intersection is less than this value, the sample is put into an unknown class else put into the class predicted by the classifier.

The accuracy of the classifier comes out to be 98%.The classifier does predict the unknown test sample into "unknown class" which seems to be impractical and finishes its working purpose in several network level objectives. As the classifier is trained on a wireless network, it is highly subjected to noise and is not suitable for live video streaming content detection as huge amount of data gets streamed in a sec.The classifier can also not resist the compression of network traffic. The training here necessitates an excessive amount of computational time and machine effort as a large size data set is employed.

Dubin et al. in [34] extended their technique of [32] on diverse browsers like Chrome, safari and Firefox etc. (browser-based streaming) founded on a passive attacker scenario. Titles are the video's general content like news, sports and education etc. The technique also exploits the variable bit rate (VBR) phenomena. The BPP feature sets are input firstly into Nearest Neighbor (NN) and then to SVM [27] algorithms. The NN algorithm is used three times. The first NN algorithm checks whether the taxonomy function calculates likenesses among test and the trained examples. This new example is then given a class according to the matching result. The NN-class algorithm calculates distinct representation to the same class examples which means that rather than keeping the whole training dataset, a whole class is denoted by a single example. The NN-class unique

procedure assumes that dataset represents an amalgamation consisting the video bursts feature sets while subtracting feature values of the unknown category class. In SVM, the features are characterized as a feature vector of the likeness score to the training title sets being used. The testing example is allocated to the class that is predicted by most of the classifiers. The testing set contains distinct title examples that are not the part of the learning set. Both algorithms calculate the likeness score among a testing title stream and training streams. The likeness score is basically the intersection of the sets of features extracted from the peaks of the streams. Maximum intersection will result in an accurate prediction of the video title. An empty set of the intersection will put the test stream in to the unknown category class.

The accuracy of the classifier is greater than 95%.The same delinquents as described for their first technique also lie with this technique.

Schuster et al. [35] established an approach grounded on convolutional neural networks (CNN) [32] to evaluate encrypted video content of video streaming sites like YouTube and Netflix etc. taking view point of a remote attacker who does not unswervingly observe the victim's stream and uses JavaScript code lines on victim's browsers for carrying out different attacks i-e cross site attack etc. It was assumed that the traffic pattern remains the same for the titles of same videos when subjected to different networks. In his own network, the attacker captures the video traces and then constructs detectors to detect the type of videos being televised on the victim's network. The largest TCP flows are firstly separated from the network traffic and then time series related features are extracted from these flows. The nine features selected here include the down bytes/sec, up bytes/sec, total bytes/sec, down packet/sec, up packet/sec, total packet/sec, down, up average and total average packet length. All the video shows are divided into 0.25 sec small chunks. The result of the former layer in the CNN signifies the matching likelihood vector. Training involves an Adadelta optimizer to adjust inputs for successive layers. The dataset comprises the titles of 11 famous TV shows and their episodes titles through different video sites such as YouTube, Netflix etc. captured by Wire shark. The normalized dataset is created by dividing each feature value by total sum value of this feature. The dataset is randomly shuffled i-e 70% is selected for learning set and remaining 30% is employed for analysis data. A distinct classifier is trained per dataset, feature and direction (incoming, outgoing and both). The forecast

of the classifier is the maximum likelihood class. A threshold is being fixed on the basis of which match accuracy is accepted from the classifier. If the maximum likelihood is above this threshold, the class match accuracy will be accepted otherwise place in to "other class" regardless of the class predicted by the classifier. Multiple features trained classifiers were also constructed. A greedy search procedure was employed on the feature set that is initiated by a feature unfilled set. After training, this set is then filled with maximum accuracy generated features. This set is then used to predict the test samples. However, the multi feature classifier failed.

YouTube classifier's accuracy per features is high for example 90% in case of only packet interval time.A distinct classifier was trained per dataset, feature and direction due to which these classifiers are not able to struggle against any change in video's compression and encryption procedures. Each feature detector should be grouped in the form of cascades in order to reduce the complexity. An efficient GPU is prerequisite as the classifier is slow to train. Data gathering is the major constraint for the learning process of the classifier and necessitates the intruder to broadcast the same stream repeatedly.

Li et al. [36] build classifiers based truly on convolutional, multilayer perceptron and recurrent neural networks to reveal the information about the streamed encrypted video's general content from different streaming sites like YouTube over a wireless network channel. The video categories used are entertainment, sports, music, comedy and style. The input to this system derived from sniffing the MAC and network layer traffic information. From MAC layer, the features extracted were the size, type, duration, properties of transmitted signals, noise rate, source and destination MAC address of the Ethernet frames. Content detection can be performed when the captured frame is a data frame. Data frames are used on the criteria that their size should be greater or equal to the minimum Ethernet size frame. They took first 3 minutes traffic of the streaming video and divide this traffic into 0.36 sec duration bins. Statistical features are then calculated from the segmented bins i-e total packets and bytes Reed et al. [40] put forward a technique to categorize the particular content titles of Netflix's videos using "kd tree" approach taking only the TCP/IP headers information with an accuracy rate of more than 99% and limited hardware over a wireless network.Adudump was utilized to calculate the size of video's application data units using both the se-

quence and acknowledgement numbers for real time network traffic and it logged the sizes for each TCP connection. This input is then fed to kd trees (where kd = 6) for constructing a classifier. The information taken from a TCP connection include the timestamps for first application data unit, most recent application data unit, successive 30 application data unit, a Boolean value to track the Netflix video flow and the title of the video that is being streamed. Once the dequeue is filled with 30 ADUs, the next incoming ADU will let the previous filled dequeuer to go through the searching process. Pearson's coefficient is being employed to find the similarity index for the testing sample.The captured streams may have retransmitted packets that can differ the searching results in most of the cases. Here the kdtrees may represent worst complexity and huge computational overhead. These may only be effective in small dimensions of data. Only the specific titles were input due to which the whole classification problem gets particular and not efficient for new data. This classifier is not applicable to the new data.in data and non-data frames. Apart from these features, four more features were selected i-e minimum, maximum, average as well as variance of packet's bulk per bin. CNN contains one input, 3 convolutional, a max pooling and 2 fully connected layers. For optimization, Adam Optimizer is employed. As number of packets varies with time, the time related features are critically to correlate. Due to this dynamic behavior, Long short-term memory network is employed as a recurrent neural network .The input is an array of size i-e time steps multiplied by a single feature (500 x 1 feature).This array is then passed as an input to the 2 fully connected layers. Every feature is consumed at a time for training the three models. The proposed model achieved different accuracies when different features are used at a time.

The accuracy of the overall classifiers is 97%.Here multilayer perceptron neural networks require high training time and an efficient GPU.All feature detectors should be group in the form of cascades to lessen the intricacy of classification. A distinct classifier is trained per feature due to which these classifiers are incapable to resist against any variation in video's compression and encryption procedures.

Dubin et al. [37] presented an unsupervised technique "k-means clustering" [38] to classify the encrypted YouTube video content traffic into general categories. The Natural Language Processing (NLP) [39] is employed with BPP sets for calculating the feature values. A BPP's integer value is transformed to a word Wi. BPP's vector space

is a set of I words combine to form a sentence.Word2vec then selects the best cluster size. The size here refers to the greatest detachment among the recent and forecasted word in a sentence. The value of the cluster is increased by one when the algorithm groups the same video streams in a single cluster. System's first component is called Preprocessor that eliminates both packet retransmissions and audio packets. The next component processes the encoded traffic to create BPPs out of video streams. Word2vec [32] is the third module that generates language from traffic features and takes BPPs to generate the new features. The fourth module is k-means algorithm that will cluster the video streams into different groups. The final module evaluates the fourth module. The captured traffic is identified into unique flows based on the IP header information such as protocol (UDP or TCP), ip addresses and port numbers of both sender and receiver. YouTube video packets are then specifically separated from other traffic by the string "googlevideos.com" found in the client-server session initiation message. Audio packets are optionally removed on the assumption that audio bursts are much smaller in size than the video bursts. All this removal occurs in the preprocessing procedure and the result is directed further.

The accuracy of the overall classifiers is 79% i-e too far to be an ideal one. The classifier requires high clustering and word transformation time as well as computation overhead. It is not suitable for live video streaming content detection as huge amount of data gets streamed in a sec.The classifier is not able to struggle against any change in video's compression and encryption procedures.

Reed et al. [40] put forward a technique to categorize the particular content titles of Netflix's videos using "kd tree" approach taking only the TCP/IP headers information with an accuracy rate of more than 99% and limited hardware over a wireless network.Adudump was utilized to calculate the size of video's application data units using both the sequence and acknowledgement numbers for real time network traffic and it logged the sizes for each TCP connection. This input is then fed to kd trees (where kd = 6) for constructing a classifier. The information taken from a TCP connection include the timestamps for first application data unit, most recent application data unit, successive 30 application data unit, a Boolean value to track the Netflix video flow and the title of the video that is being streamed. Once the dequeue is filled with 30 ADUs, the next incoming ADU will let the previous filled dequeuer to go through the searching

process. Pearson's coefficient is being employed to find the similarity index for the testing sample.The captured streams may have retransmitted packets that can differ the searching results in most of the cases. Here the kd trees may represent worst complexity and huge computational overhead. These may only be effective in small dimensions of data. Only the specific titles were input due to which the whole classification problem gets particular and not efficient for new data. This classifier is not applicable to the new data.

3.2 Comparison of Existing Encrypted Video Traffic Content Classification Techniques

Table 3.1 shows the comparison between the existing techniques describe in section 3.1.

It is evident from Table 3.1 that all the techniques employ different features and results in different outcomes. Most of the techniques are not appropriate for real-time and compression nature of the encrypted traffic. The techniques do not address the QUIC protocol traffic as this traffic is manually blocked by the researchers on a browser's settings.

Table 3.1: Existing Encrypted Video Content Detection Techniques

paper	**Dubin et al. [32]**	**Dubin et al. [34]**	**Schuster et al. [35]**	**Li et al.[36]**	**Dubin et al. [37]**	**Reed et al. [40]**
AI subset	ML	ML	DL	DL	ML	ML
Learning	supervised	supervised	supervised	supervised	unsupervised	supervised
Technique	SVM, kNN	SVM, Knn	CNN	CNN,MPN etc	k-means Clustering	kd tree
Features	flow-related	flow-related	Statistical and flow-related	Statistical and flow-related	NLP BPP words	Statistical
Content	General	General	Particular	General	General	Particular
Dataset	2700	10,000	140	3000	10,000	42,027
Streaming sites	You Tube	You Tube	YouTube etc.	You Tube	You Tube	Netflix
Streaming	Browser specific	Browser specific	Browser specific	Browser specific	Browser specific	Browser specific
Audio Removal	✓	✓	✓	✓	✓	✗
Accuracy	98%	95%	(90-98%)	97%	79%	99.5%
Error	2%	5%	(2-10%)	3%	21%	0.5%
Classification	offline	offline	offline	offline	offline	Online,offline
QUIC streaming	✗	✗	✗	✗	✗	✗
Model Updating	✗	✗	✗	✗	✗	✗
Feature Reduction	✗	✗	Greedy search ✓	✗	PCA ✓	✗
Traffic's Compression	✗	✗	✗	✗	✗	✗
Real-time network traffic	✗	✗	✗	✗	✗	✓

Chapter 4

Methodology

In this chapter, ML based encrypted video classification scheme has been proposed for monitoring users' behavior by categorizing encoded online video traffic into different classes including news, extremist videos, tv shows, etc.

4.1 ENCVIDC

Figure 4.1 demonstrates the main components of ENCVIDC:

- Data collector
- Feature Engineering
- Dataset Splitting
- Model Trainer
- Model Evaluator
- Model Upgrader
- Model Feedback

4.1.1 Data Collector

The first module deals with the gathering of encrypted video traffic. The data is collected over both available Ethernet and Wireless home network systems with Wireshark that

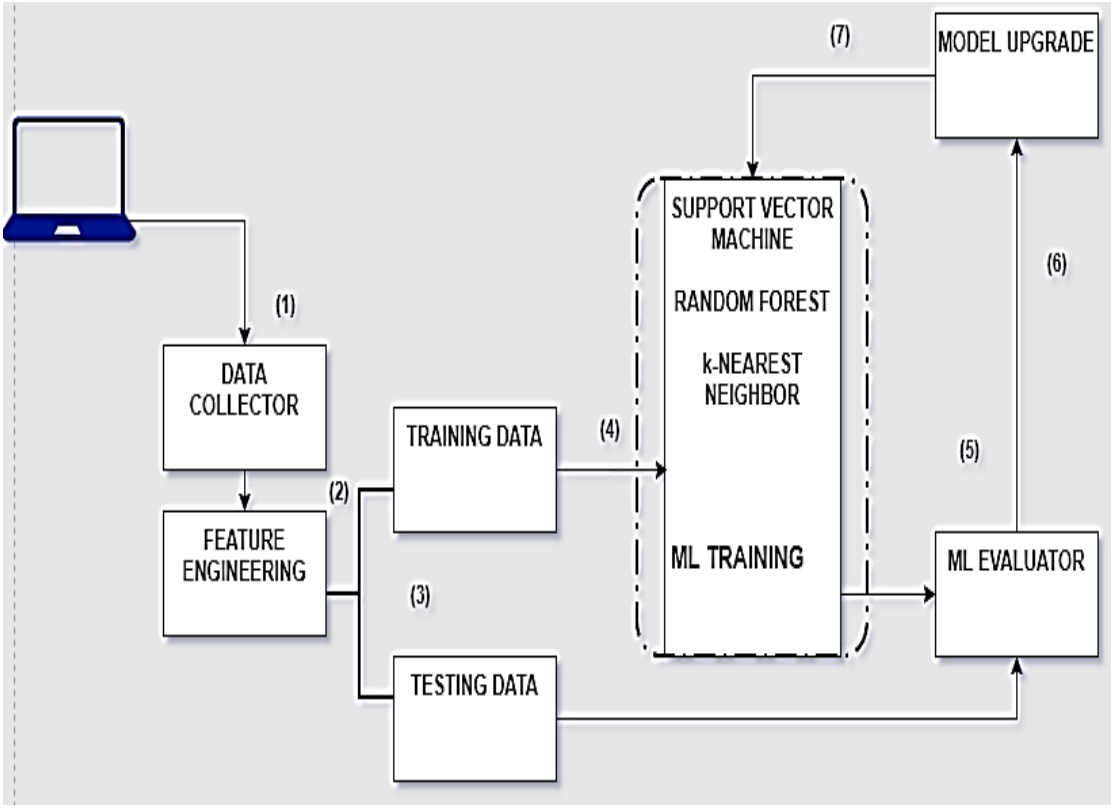


Figure 4.1: ENCVIDC

collects the samples in the form of pcap files.For this research, we have collected a total of 3036 pcap files of the different specified categories for different streaming sites i-e YouTube and daily motion etc. These files are then input one by one to the second component. The categories here presents the general nature of the content i-e sports, beauty, news and ads etc. being captured at auto 144p, 240p, 360p, 480p and 1080p quality of the video player. However, the data can be extended to further categories. Table 4.1 shows the video content classes should for this study.
From Table 4.1, blank videos are actually steady color screens. This research makes use of ad blocker to impede the ads from videos belonging to different categories. Among the collected samples, 70% of the files will be used for model's learning and the left over 30% will be utilized for validation purposes of the final constructed ENCVIDC.Table 4-2 summarizes the experimental data requirements of the new proposed system.

We have employed supervised learning algorithm that necessities the target class to be present with the sample feature values. For that purpose, we have taken the content class name as the name of the name of our pcap files as shown in figure 4.2. During parsing of packets in python, this file name will become the target class of the sample

Table 4.1: Video Content Categories

Video Content	**Streaming Sites**
Animated cartoons	YouTube,Dailymotion,Netflix,Website-specific videos
Beauty	YouTube,Dailymotion,Netflix,Website-specific videos
Blank_videos	YouTube,Dailymotion,Netflix,Website-specific videos
Cartoons	YouTube,Dailymotion,Netflix,Website-specific videos
News	YouTube,Dailymotion,Netflix,Website-specific videos
Cooking	YouTube,Dailymotion,Netflix,Website-specific videos
Educational videos	YouTube,Dailymotion,Netflix,Website-specific videos
Extremist videos	YouTube,Dailymotion,Netflix,Website-specific videos
Songs	YouTube,Dailymotion,Netflix,Website-specific videos
Sports	YouTube,Dailymotion,Netflix,Website-specific videos
TV shows	YouTube,Dailymotion,Netflix,Website-specific videos
Youtube ads	YouTube

Table 4.2: Experimental Data Requirements

Total samples	3036
Training samples	2125
Testing samples	911
Total classes	12
Samples per class	253

data. The datasets are stored as pkl files to be used for future.

4.1.2 Feature Engineering

This subsection is concerned with extracting features from the captured traffic to classify encrypted video content. The pcap samples being collected contain unique and complete TCP/UDP streams (browser specific streaming). Besides TCP, we have also taken the QUIC traffic. The streams are actually bulky data units sent from application layer to transport layer protocol. TCP divides these streams to small flows and the receiver side recombine them using sequence numbers present in the packet's headers.The detailed

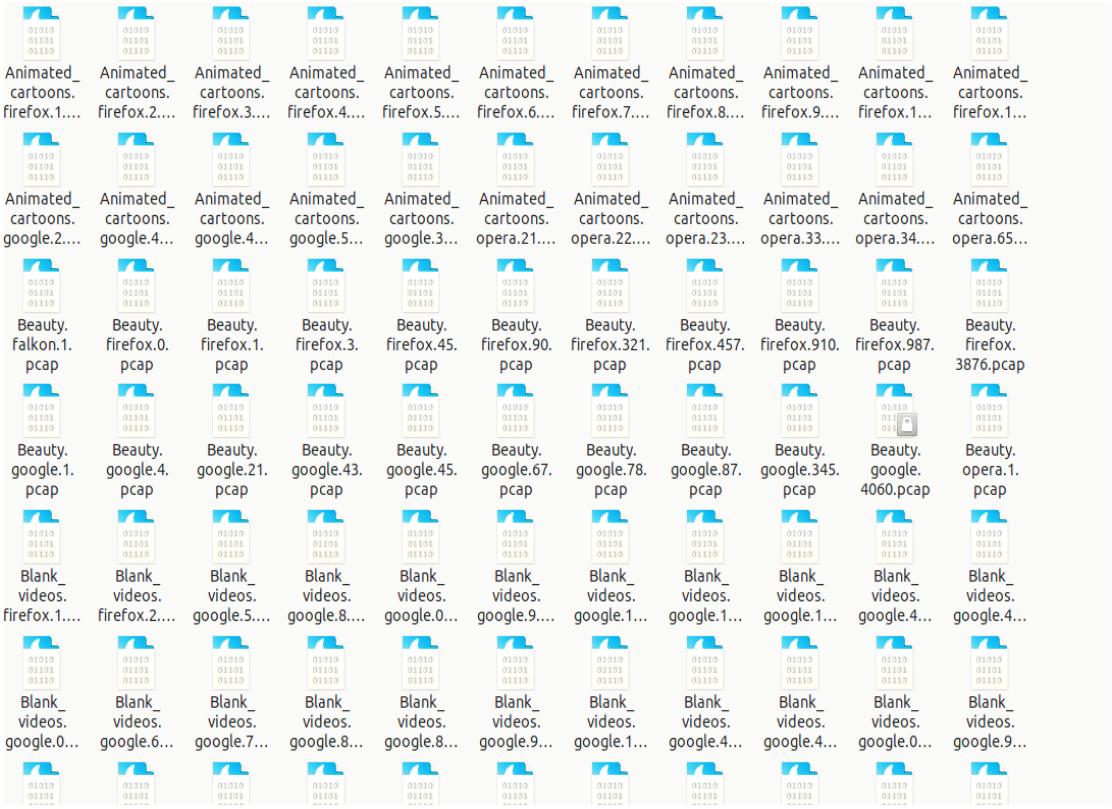


Figure 4.2: Data Samples

description of feature engineering step is given below:

Packet Retransmissions and Loss Removal:We detached both retransmitted and lost packets from the captured streams during parsing of packets on the basis of their TCP flags, sequence and acknowledgement numbers for eliminating the retransmission and loss problem. A data structure is implemented per each stream to store the corresponding sequence as well as acknowledgement numbers of the packets. If the value of sequence number of the packet is already registered, that particular packet is discarded and considered as retransmitted packet. Duplicate acknowledgement number will indicate the packet loss during the transmission. Furthermore, audio packets are not been removed grounded on the fact that the selected content classes are highly interconnected with each other.

Flow based Feature Extraction:This component firstly extracts flow-based features from the captured streams and then calculates the statistical features for the classification problem.

For the study, only the first 50 packets are taken for extracting the features. The

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	psize	psizerate	avgpsize	std(psize)	tpkts	fpkt	bpkt	fpkt/s	bpkt/s	fdata	bdata	f/bdata	b/fdata	fdata/s	bdata/s	fdata/fpkt
2	1969163	30784	820	676	2401	1421	980	22	15	1756846	86681	20	0	27465	1355	1236
3	1769390	27661	736	707	2401	1127	1274	17	19	103047	1540707	0	14	1610	24086	91
4	1969163	30784	820	676	2401	1127	1274	17	19	103047	1540707	0	14	1610	24086	91
5	1769390	27661	736	707	2401	1029	1372	16	21	264649	1628466	0	6	4137	25458	257
6	1969163	30784	820	676	2401	1029	1372	16	21	264649	1628466	0	6	4137	25458	257
7	1960343	30724	816	621	2401	1029	1372	16	21	264649	1628466	0	6	4147	25522	257
8	1769390	27661	736	707	2401	1421	980	22	15	1658993	264551	6	0	25935	4135	1167
9	1969163	30784	820	676	2401	1421	980	22	15	1658993	264551	6	0	25935	4135	1167
10	1960343	30724	816	621	2401	1421	980	22	15	1658993	264551	6	0	26001	4146	1167
11	1990772	31209	829	626	2401	1421	980	22	15	1658993	264551	6	0	26008	4147	1167
12	1792665	28025	746	778	2401	1225	1176	19	18	1580250	86779	18	0	24704	1356	1290
13	1769390	27661	736	707	2401	1225	1176	19	18	1580250	86779	18	0	24704	1356	1290
14	1969163	30784	820	676	2401	1225	1176	19	18	1580250	86779	18	0	24704	1356	1290
15	1960343	30724	816	621	2401	1225	1176	19	18	1580250	86779	18	0	24766	1360	1290
16	1990772	31209	829	626	2401	1225	1176	19	18	1580250	86779	18	0	24773	1360	1290
17	1969163	30784	820	676	2401	1421	980	22	15	1694812	148715	11	0	26495	2324	1192
18	1792665	28025	746	778	2401	1421	980	22	15	1694812	148715	11	0	26495	2324	1192

Figure 4.3: Data Samples

aim of the selection is to enable a fixed amount of packets to participate in the classification problem inorder to cut the time intricacy and bias factor. Imbalanced number of packets can favor certain classes in the classification problem.

The flow-based features may consist of header and payload based features. Classification of network traffic via port-based or payload-based analysis using static network traffic features is becoming gradually more problematic with the evolution of peer to peer applications, dynamic port numbers and encryption protocols. The precincts of the above specified analysis have motivated the use of statistical features for traffic classification such as session time etc. These dynamic traffic features are unique for every flow and enable flows to be distinguished from each other. The extraction of these features do not require high computation overhead as the structure of the flow is totally overlooked while extraction. Using the big set of flow based features, we had derived 57 significant statistical features, which are enlisted in Table 4.3 and are shown in figure 4.3.

Table 4.3: Statistical features

Number	**Features**
F1	Total, avg and std packet size
F2	Packet size per sec
F3	Total, forward and backward packets
F4	Total, forward and backward packets
F5	Total, backward and forward packets per sec
F6	Total, min and max Inter-packet time difference
F7	Average of flow duration in a stream
F8	Total,max,min and avg client data, total client data per sec
F9	Client data per backward,forward packets
F10	Total,max,min and avg server data, total server data per sec
F11	Server data per backward,forward packets
F12	Ratios of client to server,server to client data
F13	Backward and forward data
F14	Ratios of backward to forward,forward to backward data
F15	Backward,forward data per sec
F16	Backward,forward data per forward packets
F17	Backward,forward data per backward packets
F18	Total bytes,bits of data being communicated
F19	Total bytes and bits of data being communicated persec
F20	Average,forward,backward window size, forward/backward window size per second
F21	Kolmogorov's entropy for compressed data per forward packets, backward packets and second
F22	Shannon's entropy for data per forward,backward packets and second

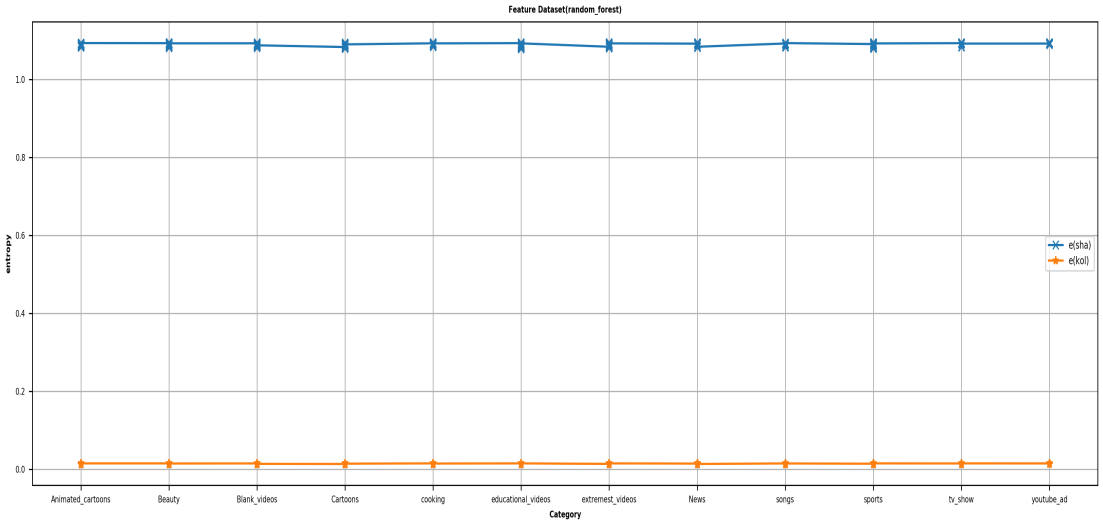


Figure 4.4: Shannon vs. Kolmogorov's Entropy

Referring to Table 4.3, Client is basically the user's browser and server is HTTP server. In video streaming scenario, the quantity of client data transmitted will be less as compared to the data delivered by the server. The forward and the backward data is truly dependent on the direction of the moving packets. Forward data is the data moving from a source to a destination whereas the backward data is the data moving from destination to the source.

For classification of the video content, we need to get into the details of the payload. As payload is encrypted, it cannot disclose any information. Nevertheless, Shannon's [41] and Kolmogorov's entropy [50] values can foresee some sort of fixed pattern in the payload. Shannon's entropy basically measures the uncertainty of data whereas Kolmogorov's entropy measures the uncertainty of compressed data. Both accurately consider the compressed and encrypted nature of network traffic.Referring figure 4.4, the values of data entropy (sha) and data entropy (kol) are actually the same in case of all the target classes and are not really alone important to be taken for classification problem.

Figure 4.5 displays how the features in Table 4.3 are correlated with each other either positively or negatively. If the correlation value is between 0 and 1, the features are correlated with each other else not related in any way. A value of 1 says that the features are highly related to each other.

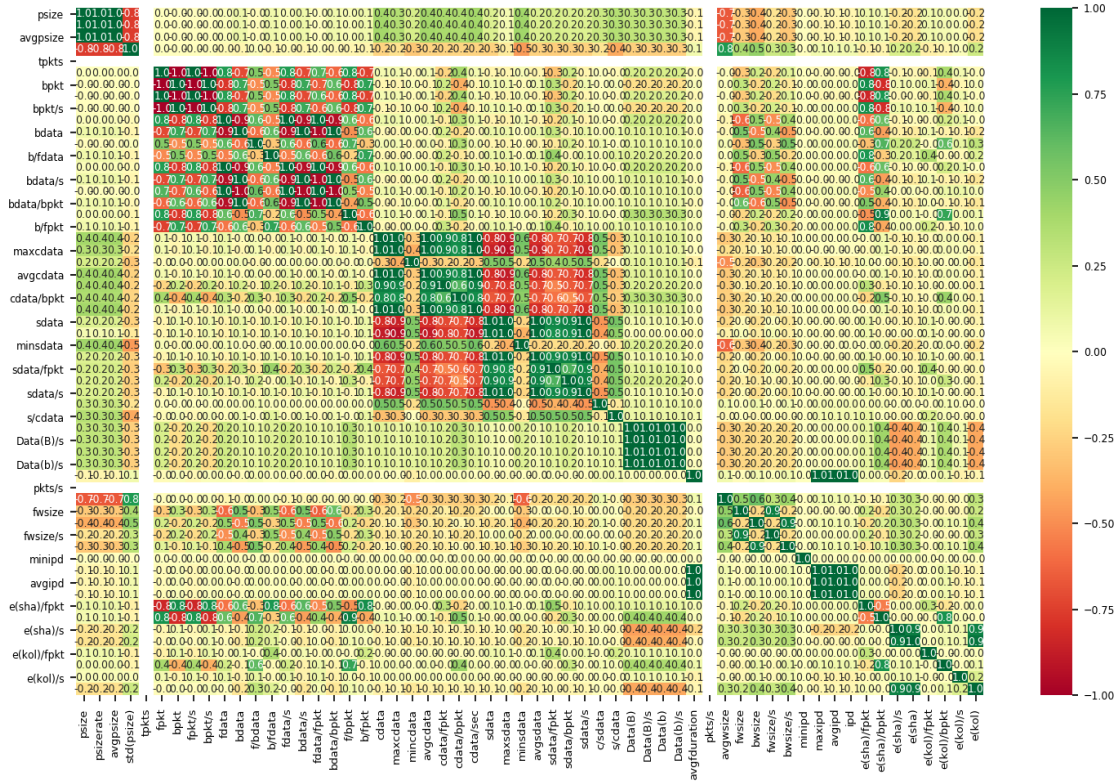


Figure 4.5: Features Correlations

Feature Reduction and Selection:It is a fact that the performance of a ML algorithm is directly related to the number of features being employed for solving the classification problem.

Assumed Features set F, the aim of feature lessening is to handpicked a features d subset such that d < F in order to minimalize the classification error.

We have considered two different cases for employing features in our stated classification problem.Case-I takes selected important features from the Table V as an input to the ML training process. The feature values are reduced not only to enhance the accuracy but also to decrease the model training, testing and updating time. Case-II, on the other hand, takes all features listed in Table 4.3 for training the ML algorithms.

There are many methods to reduce the feature set. We have employed a forward greedy search algorithm [42](presented in Table 4.4) that initially starts with an empty set and gradually fill one by one with high importance features while moving in the forward direction.We also have defined a confidence level i-e 97% as a standard threshold for accuracy for selecting the features.

Table 4.4: Forward Feature Greedy Search

Algorithm

FS0 = Ø F0 = f1,f2,.,fn;i=0,v=0,it=0;

Steps:

while(i >n)

$k = size(Fi)$;

$max\_ = 0$;

$features = 0$;

$fortfrom1tok$

$score\_ = evaluate(F(i)(j))$;

$ifscore\_ > max\_$

$max\_ = score\_; feature = F(i)(j)$;

end

end

$if(max\_ > v)$

$v = max\_$;

it = i;

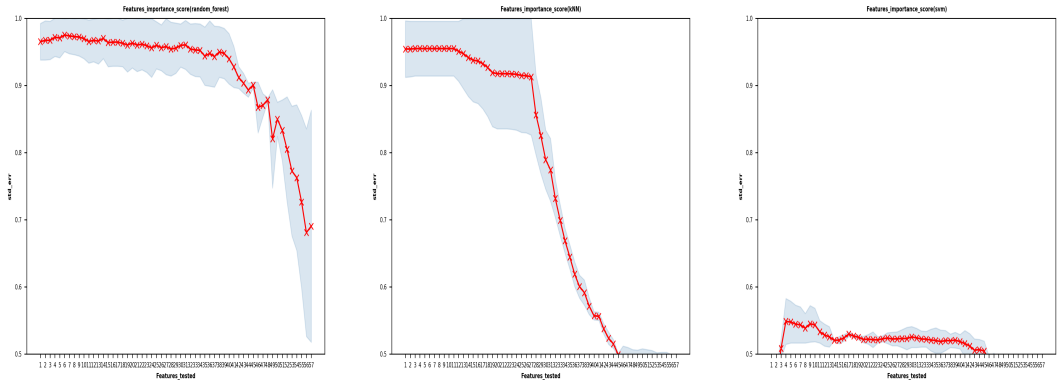
end

FS(i+1) = FSi + feature;

F(i+1) = Fi – feature;

i++;

end



(a) RF (b) kNN (c) SVM

Figure 4.6: Features Error Contribution

Fig. 4.6 a,b and c shows the corresponding error contributions of features when subjected to forward greedy search cross-validation process over respective ML techniques.

For RF, the subset of features containing the features at index 3, 39, 49, 54 i-e 'std(psize)', 'avgfduration', 'ipd' and 'e(kol)/fpkt' of the main features list are more important as they contribute to less error rates and are need to be processed further.For kNN, the subset of features containing the features at index 28, 39, 49, 56 i-e 'minsdata', 'avgfduration', 'ipd' and 'e(kol)/s' of the main features list are considered important.For SVM, the subset of features containing the features at index 0, 3, 33, 49 i-e 'psize', 'std(psize)', 'c/sdata', 'ipd' of the main features list are more important.

It should be noted that in our feature reduction case for all the three ML techniques, inter packet time duration feature is being selected. Inter packet time duration is the timestamp between two consecutive flowing packets. A video is bursty in nature. Different scenes show different arrival time i-e. An action scene can easily be detected using this time related feature. Hence, it can be considered as an important feature in our classification problem.

Feature Scaling:The reduced features will be input to the next module. The features from both the cases are to be translated in a way that they can easily be understood by the ML algorithm. Scaling basically means to change the values in such a way that they fall in a range.

Standard Scaler [45] is employed here as a feature scaler which eliminates the means and adapt to unit variances i-e shape the data into a range using the Gaussian

```
Input: [-0.04987336 -0.05090828 -0.05353102  1.51867142  0.        -0.52211309
  0.52211309 -0.57245784  0.38705631  0.50450224 -0.86475471  0.55357148
 -0.41843699  0.50428323 -0.86485083  0.9135703  -0.95502113 -0.84196844
  0.88917334  0.76464517  0.85445717 -0.41379698  0.76464574  0.81602945
  0.37582616  0.76438285 -1.04869076 -0.95706766 -0.54280424 -1.04869219
 -0.84515714 -0.95842857 -1.04883867  0.71073981 -0.26344972 -0.94645693
 -0.94755985 -0.94645693 -0.94760612  0.13005919  0.         0.77729619
 -0.38377293  1.36473997 -0.36735987  1.78855229 -1.         0.13005919
  0.13005919  0.1107693   0.36334495 -0.26801789  0.69665331  0.71087411
 -0.01815184 -0.08926437 -0.08737041]
```

Figure 4.7: Scaled Features

distribution. Firstly the data will fit and then transform to a particular format. This standardizes the data into a standard normal distribution as shown in figure 4.7.

The scaled features are now ready to be trained by a ML algorithm (RF can work without the scaler). We have employed "StandardScaler" function from scikit-learn package "data preprocessing".Once the features are scaled, the dataset is then distributed to 70% training and 30% testing subsets as shown in step 3 of figure 4.1. The training set is now organized to be fed into three classical ML algorithms.The formula used here for standardization is as under:

$$Z = (x - \mu)/\sigma \tag{4.1.1}$$

Here x are feature values, µ is the mean of observed features values and σ represents the standard deviation.The mean and standard deviation is calculated by the subsequent formulas:

$$\mu = \sum(x)/total\_count(x) \tag{4.1.2}$$

$$\sigma = \sqrt{(\sum(x - \mu)^2)/total\_count(x))} \tag{4.1.3}$$

We will normalize each column so that the value of µ becomes 0 and σ becomes 1.

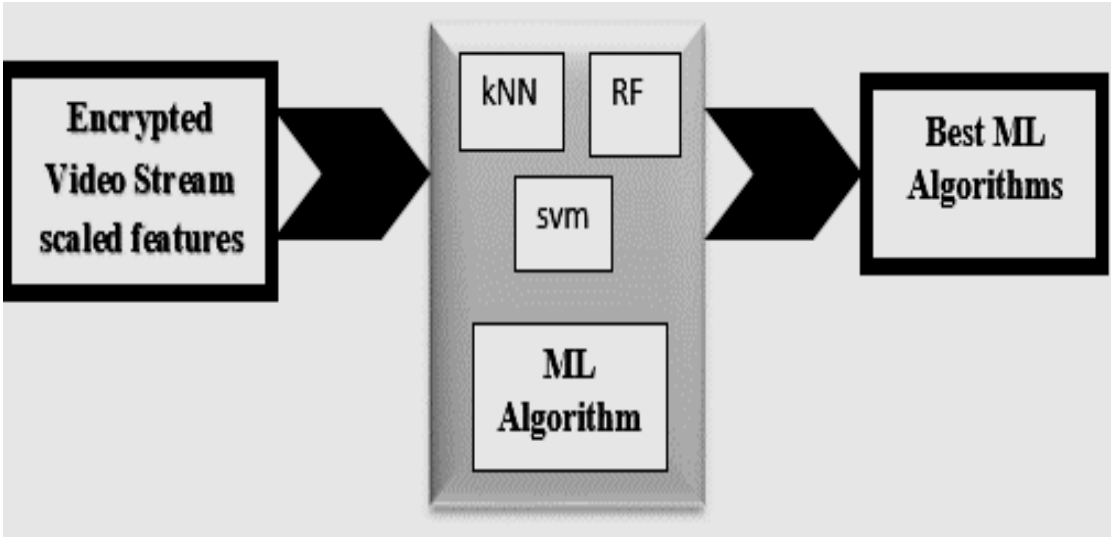


Figure 4.8: Model Selection

4.1.3 Model Training

The ML classifiers selected here are ensemble RF, SVM and kNN. The superlative ML technique will be nominated on the basis of performance attributes and metrics after training as well as testing the models against the data at the end.The purpose of using the model selection is to assess which particular type of ML technique is best suited to the encrypted video content classification as the content categories are closely related to each other.Employing a model according to one's own choice cannot be effective in any case regardless of the feature values being chosen. Figure 4.8 shows the selection of best model for ENCVID.

Model Tuning:We need to tune models by their hyper parameters in order to fit them for the training data. The selection of hyper parameters is done here by "grid search" [42] method. Grid search consists of the following components:

- A classifier or an estimator (RF,kNN,SVM)
- A parameters grid/dictionary (dictionary of required parameters)
- A scheme for probing and grouping samples (Sequential or Parallel Pipeline)
- Cross validation scheme(3-validation method)
- A "score function"(accuracy)

Table 4.5: Grid search

Algorithm

Input: Hyper parameters HRF,HkNN,Hsvm

Output: value = best hyper parameters key

Estimators: RF,SVM,kNN

Import dataset T

$for\ i\ values\ of\ each\ key\ HRF,\ HkNN,\ Hsvm$

$do:$

$for\ 3-cross\ validation\ by\ each\ estimator$

$do:$

$calculate\ the\ maximum\ accuracy\ value\ Amax$

end

end

The parameters grid/dictionary is used through a Pipeline that checks each combination in a parallel manner in order to check maximum parameters combinations, minimize the tuning time of the model and processing time of the processors. This is done by the keyword "n-jobs = -1" in the parameter list of the grid search.It means that all the available processors of the system will be engaged in the process.

We have defined certain ranges for each of the parameter according to our requirements. Default model settings cannot output the desired results and can lead to misclassifications.The algorithm for grid search for all the classifiers is shown in Table 4.5.

This search is performed by "GridSearchCV" function from sklearn package "model selection".We have employed 3-fold cross validation method for the division of whole training dataset to small datasets and used them as training and testing sets for creating a model prototype for choosing the optimized hyper parameters.The general procedure followed is as under:

- We shuffled the training dataset.
- We then split the training set into 3 equal size datasets.
- Each unique dataset is then divided into train and test datasets.

- We train the models on training set and then evaluate them against the test dataset.
- We then summarize the results on the basis of our performance metric i-e accuracy.

Table 4.6,7,8 shows the optimized hyper parameters for ML techniques with reference to both the cases described above.

Table 4.6: Optimized Hyper-parameters for RF

Hyperparameters	**Case-I**	**Case-II**
Bootstrap	True	False
Criterion	Entropy	Entropy
Max_features	Auto	Sqrt
Min_samples_leaf	02	02
Min_samples_split	07	08
Estimators	28	15

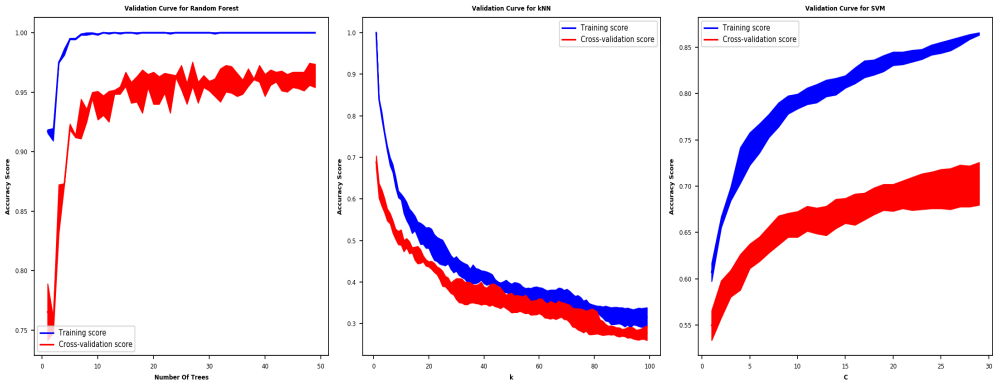
Table 4.7: Optimized Hyper-parameters for kNN

Hyperparameters	**Case-I**	**Case-II**
Algorithm	Auto	Auto
Metric	Euclidean	Manhattan
Neighbors	17	01
Weights	Distance	Uniform

Table 4.8: Optimized Hyper-parameters for SVM

Hyperparameters	**Case-I**	**Case-II**
C	26826.95	100000.0
γ	10.0	0.001
Kernel	rbf	rbf

Choosing the right hyper parameters is critical to a classification problem for in-



(a) RF (b) kNN (c) SVM

Figure 4.9: Optimized Hyperparameters vs. Accuracy

stance Fig. 4.9a shows the behavior of a RF model when number of trees in a forest is increased.It is evident that the training scores become constant when the number of DTs in RF exceed from 20 and it is desirable to take minimum number of trees to avoid complexity. Fig. 4.9b shows the behavior of a kNN model when number of neighbors are altered.It becomes visible that with the increase in the number of neighbors in kNN both training and validation scores decreases.Fig. 4.10c shows the behavior of a SVM model when the cost factor is being altered.With the increase in cost factor and decreasing gamma value in SVM, both training and validation scores increases.

Once the models are tuned, the training process starts.During the training phase, the models will get themselves trained using the training dataset. An optimized threshold is calculated by using the cross validation process and the AUC-ROC score. This threshold value will serve the purpose for an accurate classification process. The threshold value is calculated as an average of maximum threshold of each class as shown in figure 4.11.

For RF, the optimized threshold comes out to be 0.308 and 0.333 for case-I and II respectively. For kNN, the optimized threshold comes out to be same i-e 0.4 for case-I and II respectively. For SVM, the optimized threshold comes out to be same i-e 0.675 for case-I and II respectively.

4.1.4 Model Evaluator

This component will load the trained models and evaluates them against the testing data by using the performance metrics. Besides the accuracy performance of classifiers,

```
y_predict_proba = model.predict_proba(X)
n_classes = len(labels)
global optimal_idx,threshold_
all_y_test_i = np.array([])
all_y_predict_proba = np.array([])
for i in range(n_classes):
        y_test_i = map(lambda x: 1 if x == i else 0, Y)
        all_y_test_i = np.concatenate([all_y_test_i, y_test_i])
        all_y_predict_proba = np.concatenate([all_y_predict_proba, y_predict_proba[:, i]])
        fpr[i], tpr[i], threshold[i] = roc_curve(y_test_i, y_predict_proba[:, i])
        roc_auc[i] = auc(fpr[i], tpr[i])
        optimal_idx = np.argmax(tpr[i] - fpr[i])
optimal_threshold = threshold[optimal_idx]
norm = [float(i)/sum(optimal_threshold) for i in optimal_threshold]
threshold_ = sum(norm)/len(norm)
```

Figure 4.10: Threshold

the time-related performance metrics will also be evaluated.During this phase, a testing threshold is calculated that is matched against the optimized threshold. If the threshold value is less than or equal to optimize threshold, the model will not update and in the opposite case, the model gets itself trained with the new data.

4.1.5 Model Upgrade

This module focused on upgrading the model based on threshold value. A threshold value is calculated for each class during the testing and the training phase. If the testing threshold is above the optimal threshold, the model is upgraded with the new data record else discarded as shown as step 7 in figure 4.1. In the other case, irrespective of the class predicted by the classifier, the prediction of the model will not be correct and the model will not be updated with this sample.

For RF, the testing threshold comes out to be 0.364 and 1.00 for case-I and II respectively. For kNN, the testing threshold comes out to be same i-e 0.4 for case-I and II respectively. For SVM, the testing threshold comes out to be 0.679 for case-I and 0.678 for case-II respectively.

Table 4.9: Experimental Setup

Attributes	**Requirements**
Processor	Intel® CoreTM i7-4790 CPU@3.60GHz x4
Operating system	Ubuntu 18.04.5
Programming Language	python 3.6
Sniffer	Wireshark
ML library	Scikit-learn 0.20.0
Data Analysis Libraries	Pandas 1.1.4,Numpy 1.19.0
Data Visualization Library	MATLAB R2019b
Python package installer	pip 20.2.4

If the trained model will be subjected to some unknown test class (labels other than the specified ones) then firstly that would be classified to a more closely related class. If this sample is not closely related to any of the target class, the model will wait for at least two records for that label, train itself and then test them. This component serves to be a feedback to the model to train on new data.

4.2 Experimental Setup

We have employed most common browsers for windows and Linux to capture the video streams of variable length at different time intervals in pcap files through Wireshark. These pcap files are then parsed one by one by a python script parser to extract the features in the form of a csv file. The pcaps are deleted permanently to free space once they are parsed. These files are then divided to two sub csv files. The training csv files are then utilized by each of the ML algorithm one by one to create the respective train models. The models are created by employing the most common python based ML library Scikit-learn. These trained models are stored as jbl file for testing the data. A python script then loads all the models at a time and display the outcomes. The hardware and software requirements of our scheme is enclosed in Table 4.13.

Chapter 5

Discussion and Analysis

This chapter illustrates the analysis of the proposed method ENCVIDC.

5.1 ML Model Evaluation

The performance and effectiveness of the ML models are evaluated by following parameters as accuracy alone cannot determine the performance of the classifiers.

5.1.1 Classification Report

This includes the scores for precision, recall, f1-score and support of the testing data against the employed ML techniques.

Precision represents a score about how often the models predicts the correct video content class. Recall represents the percentage of positive samples video content classes correctly predicted by the models. F1-score represents the harmonic means of precision and recall scores.All the above parameters are calculated by using the following equations:

Precision

$$(TP/(TP + FP)) \tag{5.1.1}$$

Recall

$$(TP/(TP + FN)) \tag{5.1.2}$$

F1-score

$$(TP/1/2(FP + FN)) \tag{5.1.3}$$

	precision	recall	f1-score	support
Animated_cartoons	0.99	0.97	0.98	76
Beauty	1.00	1.00	1.00	76
Blank_videos	0.99	0.99	0.99	76
Cartoons	0.93	0.99	0.96	76
News	0.96	0.99	0.97	76
cooking	1.00	1.00	1.00	76
educational_videos	0.97	1.00	0.99	76
extremest_videos	1.00	1.00	1.00	75
songs	0.99	0.93	0.96	76
sports	0.97	0.96	0.97	76
tv_show	0.99	0.96	0.97	76
youtube_ad	0.99	0.97	0.98	76
micro avg	0.98	0.98	0.98	911
macro avg	0.98	0.98	0.98	911
weighted avg	0.98	0.98	0.98	911

Figure 5.1: Classification Report (RF Case-I)

Here TP, FP, FN and FP represent true positive, false positive, false negative and false positive negative. These parameters are used for our model to determine the quality of classification performed by our classifiers as the accuracy rate cannot exclusively decide the extent to which the classes can be classified.

The higher values of sensitivity means there are large number of true positives as compared to false negatives. It is evident that some of the video content classes for instance beauty videos are predicted with maximum accuracy rate by all ML techniques.

Figures 5.1, 5.2 and 5.3 encloses the classification reports of the ML models. Figure 5.4 show the average classification report metric scores against each ML case respectively. Here 1 represents the highest score and 0 represents the lowest score.

Referring to the figure 5.7, the classification report metrics for RF are higher in values as compared to the rest of the ML classifiers. SVM is considered to be least accurate in our classification problem.

5.1.2 Confusion Matrix

Confusion Matrix plots the guessing probabilities of samples against all the possible classes by a table representing the TP, TN, FP and FP values of the predictions to visualize the performance of the ML algorithms. We basically have two dimensions here i-e. Actual content and predicted content. As we have employed supervised ML, actual content type is known and predicted content will be determined the ML algorithm.

	precision	recall	f1-score	support
Animated_cartoons	0.97	0.99	0.98	76
Beauty	1.00	1.00	1.00	76
Blank_videos	1.00	1.00	1.00	76
Cartoons	1.00	0.99	0.99	76
News	0.89	0.96	0.92	76
cooking	1.00	1.00	1.00	76
educational_videos	0.99	0.91	0.95	76
extremest_videos	1.00	1.00	1.00	75
songs	1.00	0.97	0.99	76
sports	0.96	0.97	0.97	76
tv_show	0.99	1.00	0.99	76
youtube_ad	0.99	0.99	0.99	76
micro avg	0.98	0.98	0.98	911
macro avg	0.98	0.98	0.98	911
weighted avg	0.98	0.98	0.98	911

Figure 5.2: Classification Report (RF Case-II)

	precision	recall	f1-score	support
Animated_cartoons	0.58	0.70	0.63	76
Beauty	0.93	0.83	0.88	76
Blank_videos	0.78	0.62	0.69	76
Cartoons	0.65	0.82	0.73	76
News	0.61	0.58	0.59	76
cooking	0.71	0.88	0.79	76
educational_videos	0.57	0.42	0.48	76
extremest_videos	0.64	0.61	0.63	75
songs	0.57	0.51	0.54	76
sports	0.41	0.33	0.36	76
tv_show	0.72	0.84	0.78	76
youtube_ad	0.73	0.80	0.76	76
micro avg	0.66	0.66	0.66	911
macro avg	0.66	0.66	0.66	911
weighted avg	0.66	0.66	0.66	911

Figure 5.3: Classification Report (kNN Case-I)

	precision	recall	f1-score	support
Animated_cartoons	0.58	0.70	0.63	76
Beauty	0.93	0.83	0.88	76
Blank_videos	0.78	0.62	0.69	76
Cartoons	0.65	0.82	0.73	76
News	0.61	0.58	0.59	76
cooking	0.71	0.88	0.79	76
educational_videos	0.57	0.42	0.48	76
extremest_videos	0.64	0.61	0.63	75
songs	0.57	0.51	0.54	76
sports	0.41	0.33	0.36	76
tv_show	0.72	0.84	0.78	76
youtube_ad	0.73	0.80	0.76	76
micro avg	0.66	0.66	0.66	911
macro avg	0.66	0.66	0.66	911
weighted avg	0.66	0.66	0.66	911

Figure 5.4: Classification Report (kNN Case-II)

	precision	recall	f1-score	support
Animated_cartoons	0.48	0.51	0.50	76
Beauty	1.00	0.97	0.99	76
Blank_videos	0.67	0.71	0.69	76
Cartoons	0.54	0.63	0.58	76
News	0.55	0.58	0.56	76
cooking	0.58	0.80	0.67	76
educational_videos	0.53	0.34	0.42	76
extremest_videos	0.50	0.47	0.48	75
songs	0.74	0.46	0.57	76
sports	0.38	0.26	0.31	76
tv_show	0.65	0.80	0.72	76
youtube_ad	0.52	0.61	0.56	76
micro avg	0.60	0.60	0.60	911
macro avg	0.60	0.60	0.59	911
weighted avg	0.60	0.60	0.59	911

Figure 5.5: Classification Report (SVM Case-I)

	precision	recall	f1-score	support
Animated_cartoons	0.48	0.51	0.50	76
Beauty	1.00	0.97	0.99	76
Blank_videos	0.67	0.71	0.69	76
Cartoons	0.54	0.63	0.58	76
News	0.55	0.58	0.56	76
cooking	0.58	0.80	0.67	76
educational_videos	0.53	0.34	0.42	76
extremest_videos	0.50	0.47	0.48	75
songs	0.74	0.46	0.57	76
sports	0.38	0.26	0.31	76
tv_show	0.65	0.80	0.72	76
youtube_ad	0.52	0.61	0.56	76
micro avg	0.60	0.60	0.60	911
macro avg	0.60	0.60	0.59	911
weighted avg	0.60	0.60	0.59	911

Figure 5.6: Classification Report (SVM Case-II)

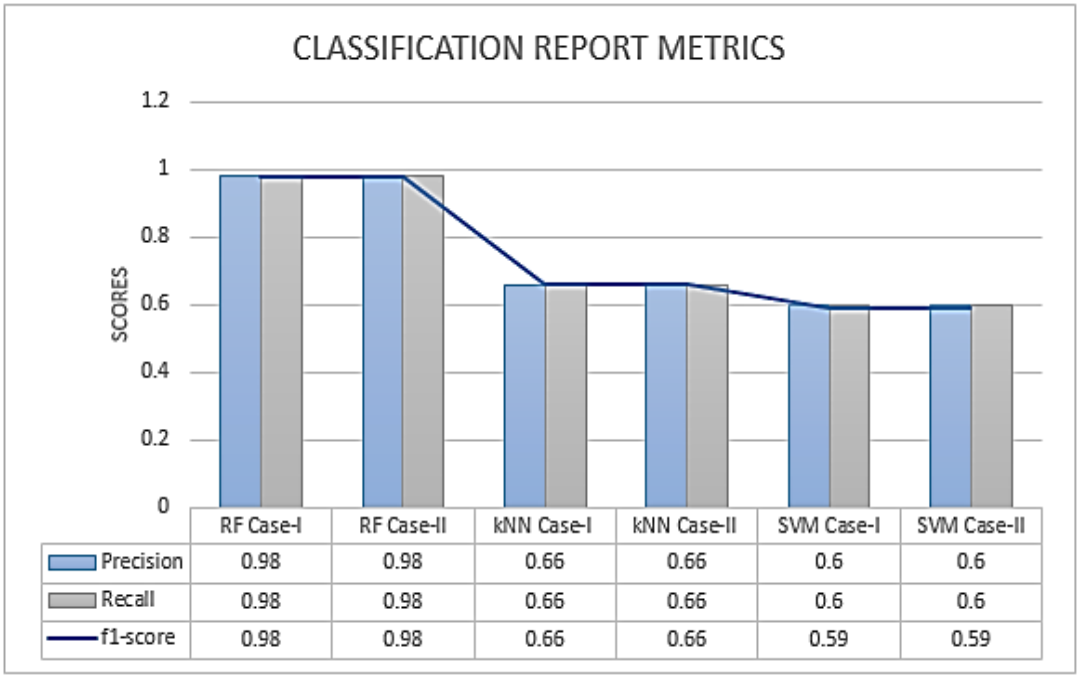


Figure 5.7: Comparison of Classification Report Metrics

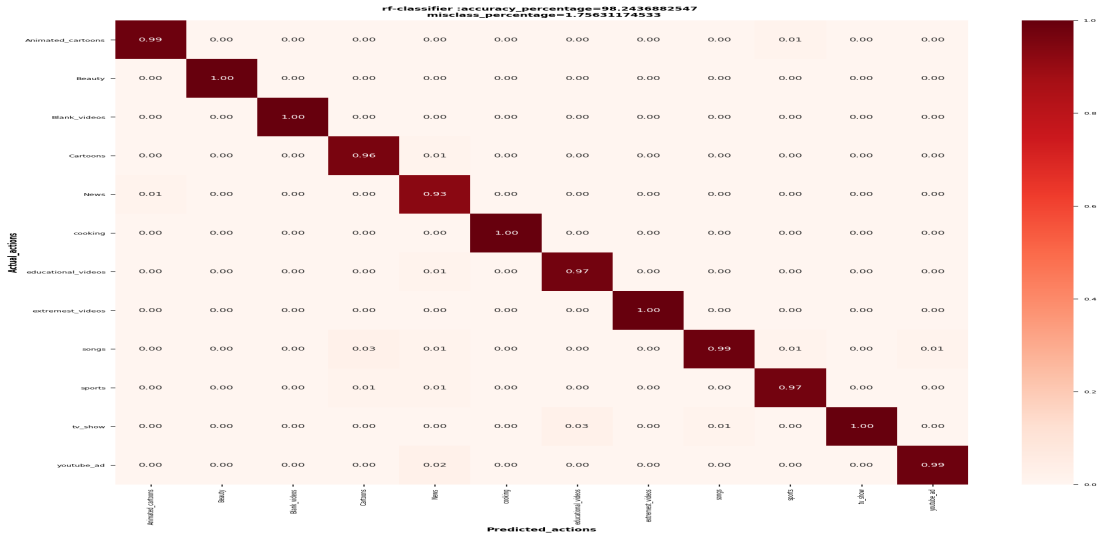


Figure 5.8: Confusion Matrix (RF Case-I)

The probability of "1" shows that the video content class is highly accurately identified by the classifier and "0" means that the predicted class is not the actual class. The values other than 0 or 1 are the misclassifications of the models for the test sample. There may be misclassifications in our scenario due to the highly interconnected nature of the videos for instance a YouTube ad may contain cartoons or music in it.

With the help of confusion matrixes 5.8-5.13, graph in figure 5.14 presents the predicting behaviors of ML algorithms. Here we can observe that most of the classes are predicted with maximum accuracy whereas some closely related classes have shown some error rates as well for instance TV shows and songs. Songs can be a part of a TV show and a YouTube ad.

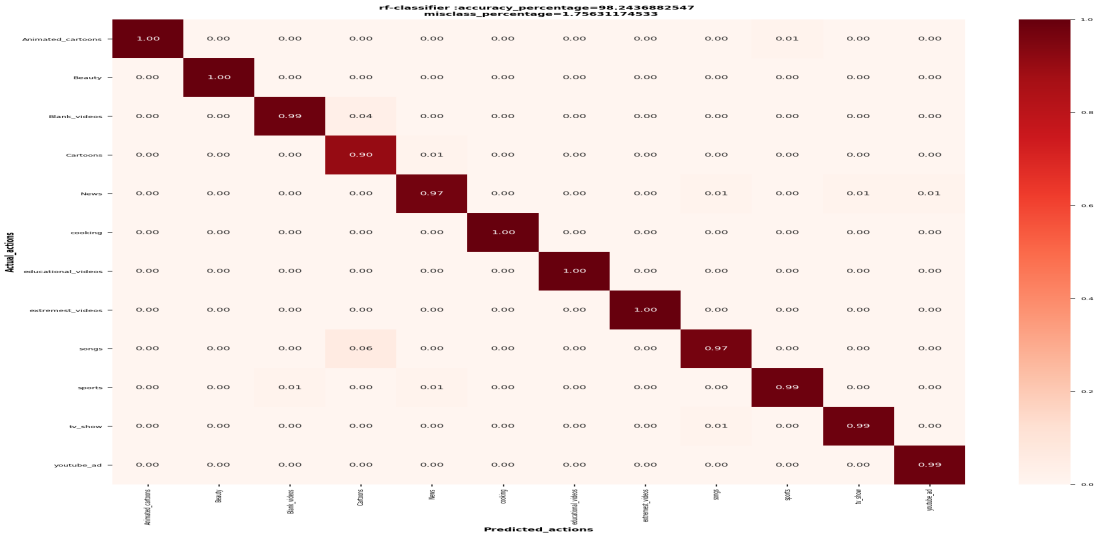


Figure 5.9: Confusion Matrix (RF Case-II)

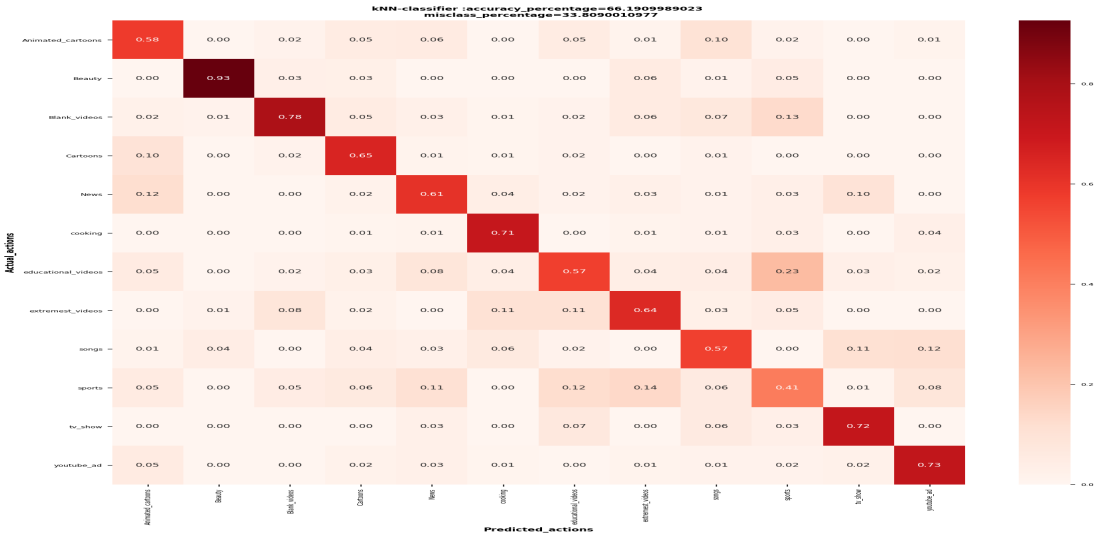


Figure 5.10: Confusion Matrix (kNN Case-I)

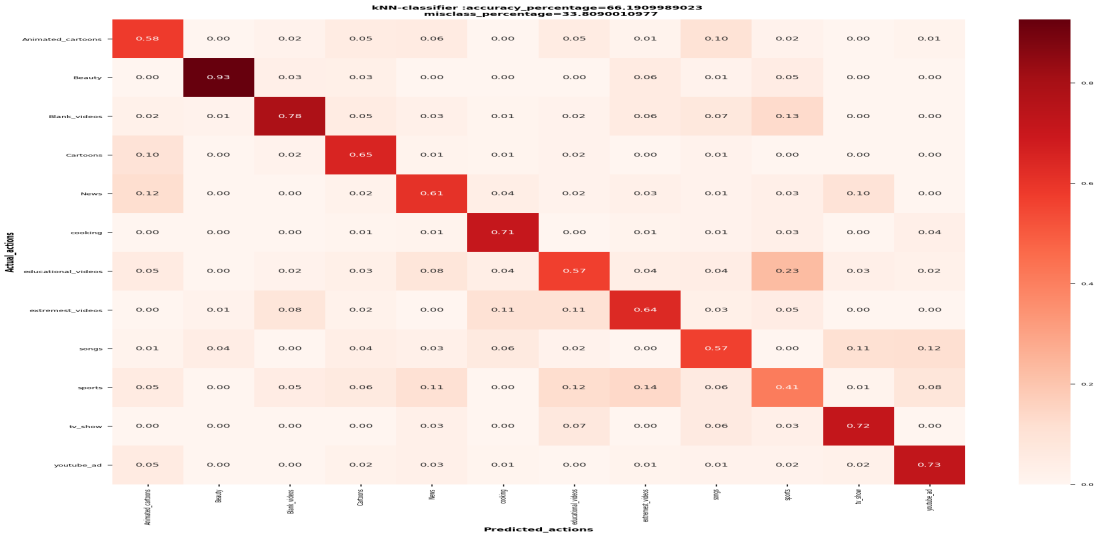


Figure 5.11: Confusion Matrix (kNN Case-II)

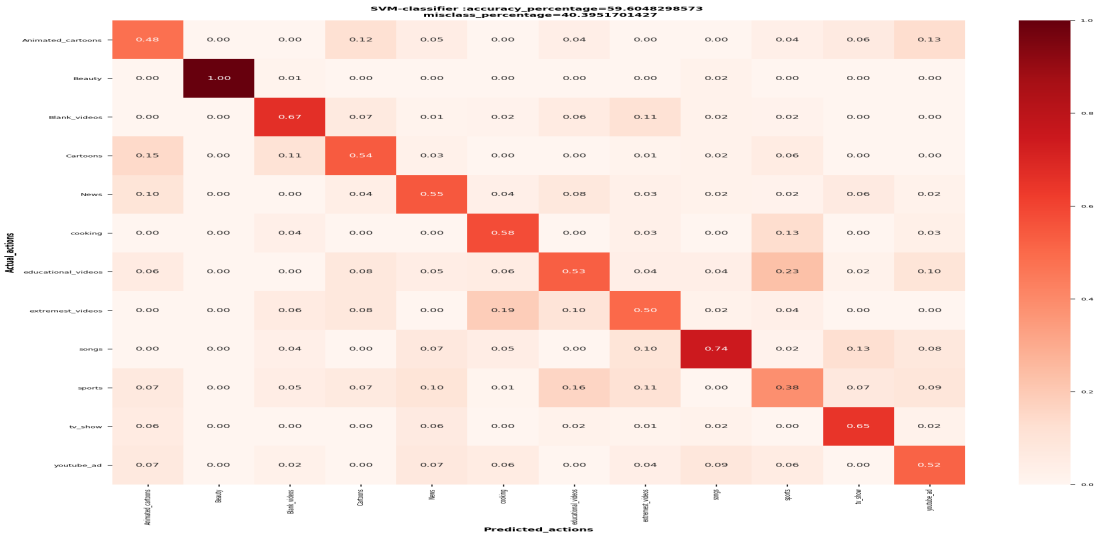


Figure 5.12: Confusion Matrix (SVM Case-I)

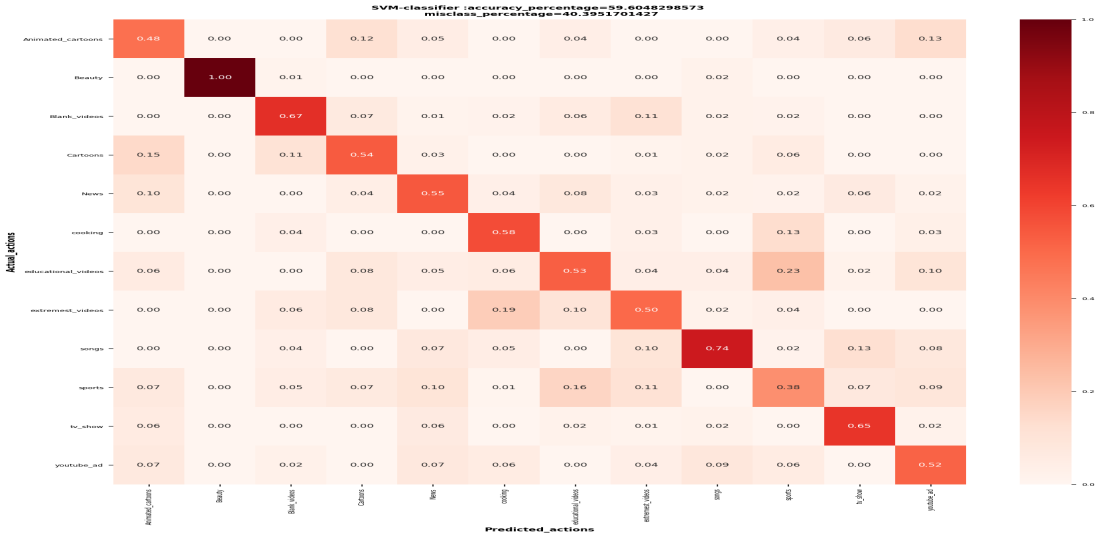


Figure 5.13: Confusion Matrix (SVM Case-II)

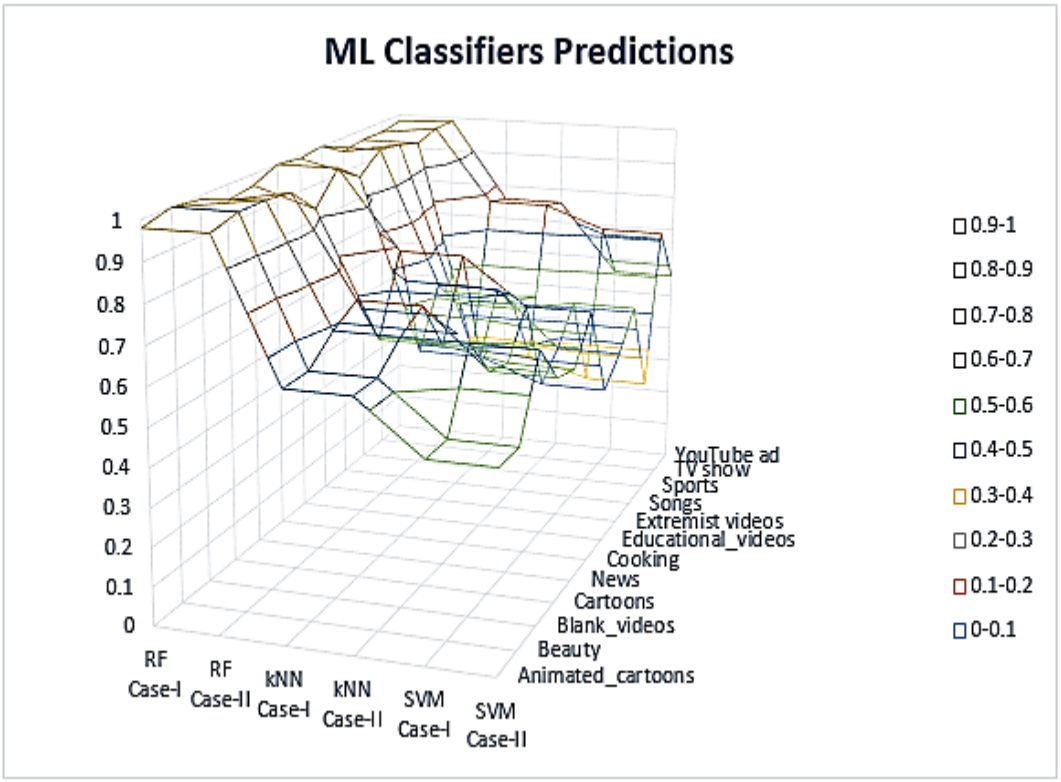


Figure 5.14: Prediction Probabilities

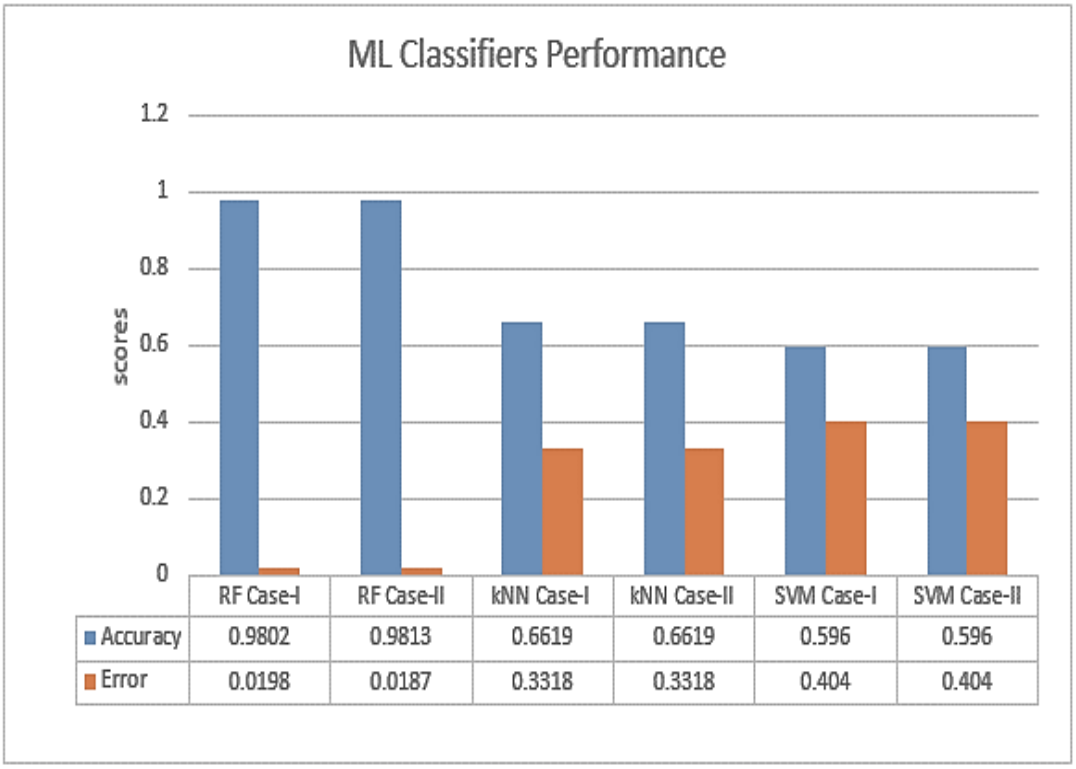


Figure 5.15: ML Classifiers Performance

The graph in figure 5.14 is showing the predicted probabilities against the actual classes for the ML classifiers .The allocation of the specific class is done by comparing this probability with the calculated optimized threshold.

We have got almost negligible error rate in case of RF and maximum in case of SVM.

5.1.3 Accuracy

Figure 5.15,shows the accuracy and error report of the ML classifiers when subjected to the testing data. The formula used here to calculate the accuracy is as under:

$$Accuracy = TP + TN/(TP + FP + TN + TP) \tag{5.1.4}$$

It is evident that Random Forest case II is more accurate as compared to the other candidates and contribute least to the error rate.

Table 5.1: Comparison of ML Classifiers

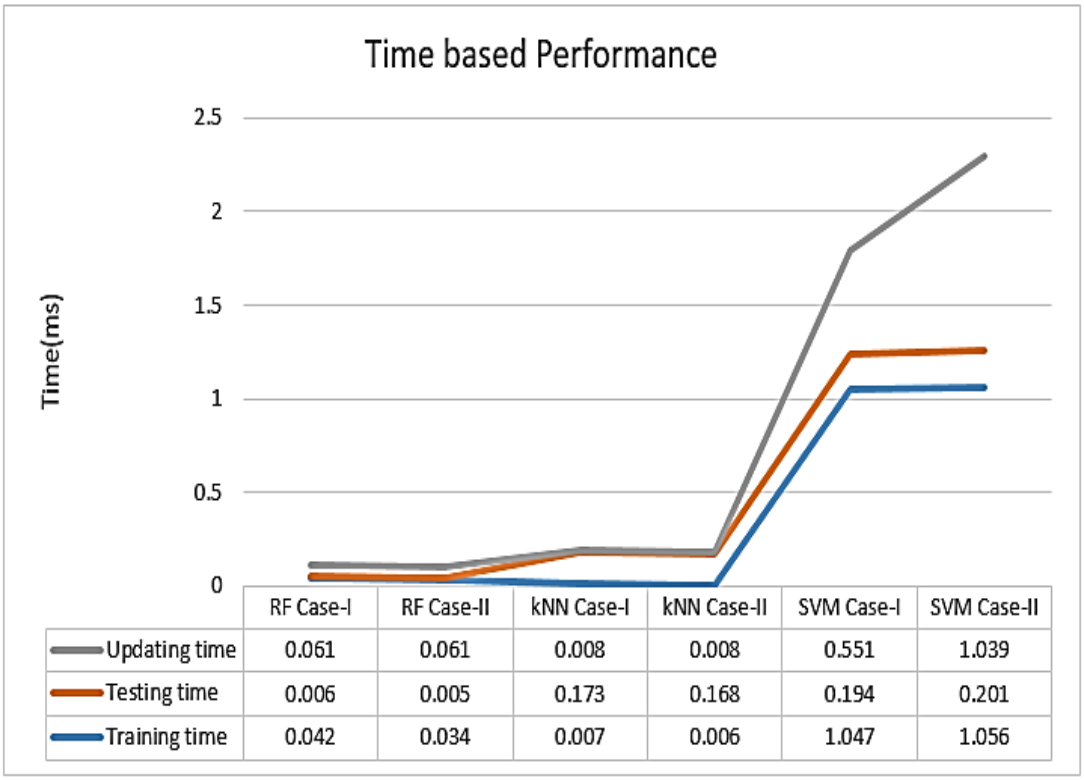
Attributes	**RF Case-I**	**RF Case-II**	**kNN Case-I**	**kNN Case-II**	**SVM Case-I**	**SVM Case-II**
Training time	0.089 sec	0.073 sec	0.016 sec	0.013 sec	2.225 sec	2.244 sec
Training time/Record	0.042 ms	0.034 ms	0.007 ms	0.006 ms	1.047 ms	1.056 ms
Training accuracy	99%	89.9%	95%	62%	55%	61%
Testing time	0.005 sec	0.004 sec	0.157 sec	0.153 sec	0.177 sec	0.183 sec
Testing time/Record	0.006 ms	0.005 ms	0.173 ms	0.168 ms	0.194 ms	0.201 ms
Testing accuracy	98.24%	98.13%	66.19%	66.19%	59.6%	59.60%
Error rate	1.98%	1.87%	33.81%	33.81%	40.4%	40.4%
Updating time	0.055 sec	0.055 sec	0.007 sec	0.007 sec	0.501 sec	0.946 sec
Updating time/Record	0.061 ms	0.061 ms	0.008 ms	0.008 ms	0.551 ms	1.039 ms

5.2 Model Selection

As a lot of network traffic streamed in a second, we need such a technique that is both time efficient and accurate. For offline content detection, accuracy is the most important factor since no traffic is streamed whereas in a real-time scenario, both time and accuracy are considered important. For our case, the new scheme should be time efficient and accurate to fulfill the security requirements in an effective manner. Table 5.2 summarizes our experiment for finding the best ML algorithm to drive ENCVIDC.All the algorithms are given the same conditions and data samples.

If the testing scores are less than the training scores, the created model is not effective whereas in the opposite case i-e training score less than validation score leads towards an effective model.

Referring to Table XI, we have discarded the models RF Case-I,kNN Case-I, SVM Case-II for serving as the leaner in ENCVIDC as in these cases the models are getting



	RF Case-I	RF Case-II	kNN Case-I	kNN Case-II	SVM Case-I	SVM Case-II
Updating time	0.061	0.061	0.008	0.008	0.551	1.039
Testing time	0.006	0.005	0.173	0.168	0.194	0.201
Training time	0.042	0.034	0.007	0.006	1.047	1.056

Figure 5.16: ML Classifiers time-based Performance

over fitted with data and outliers.In simple words, these models are more trained but poorly detecting the new data samples.RF case-II IS giving promising results.Although kNN Case-II is more time efficient but it is far away to be ideal in our case. SVM case-I took somewhat more training time and computation power as compared to the rest of the cases. They are far behind to be an ideal and accurate classifier.Figure 5.16 represents the time based performance of the employed classifiers against a single data sample.

The classifier, we selected produces a low error while testing the original training set and testing set. The classifier is a right balance for “bias” and “variance” towards the testing sample. Results may differ when huge amount of data is being used but RF Case-II will always stand out.

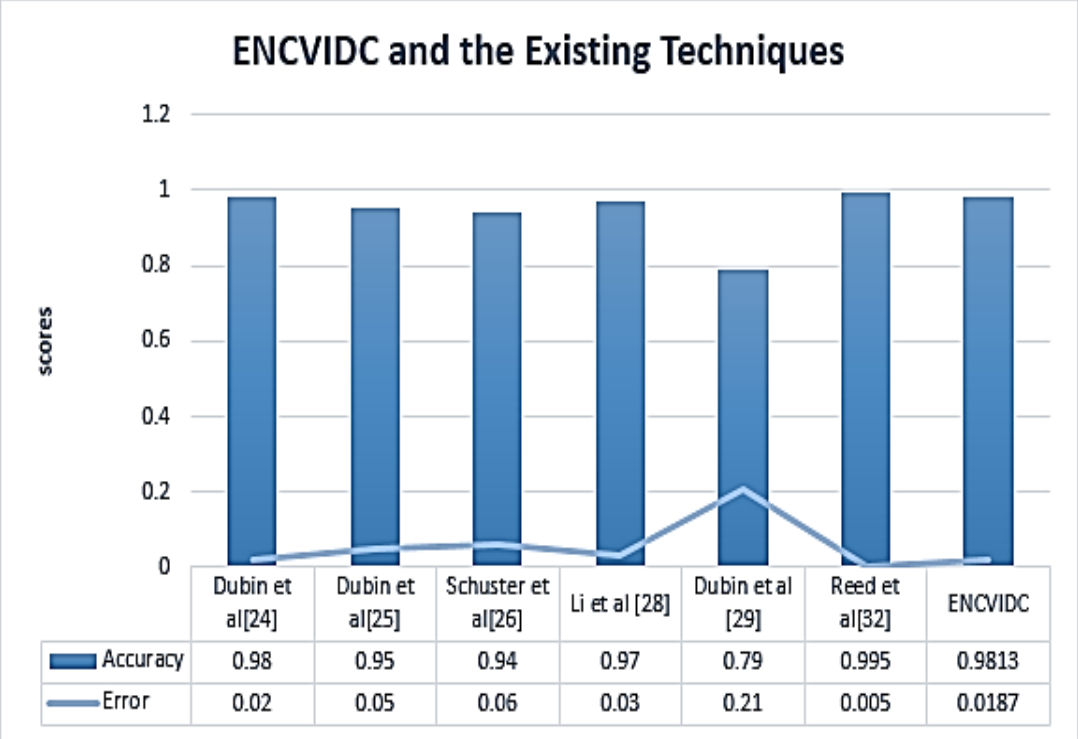


Figure 5.17: Comparison of ENCVIDC with Existing Techniques

5.3 Comparison with the Existing Techniques

Table 5.3 displays the contrast of our new technique with the existing encrypted video content classification techniques.

ENCVIDC has presented the feature of upgrading the already trained model with the true predicted test samples data on the basis that the model should remain updated with the new data. The existing techniques as described in Table 3.1 lack this facility. These also classify the new test sample (other than the specified) to "unknown" classes but ENCVIDC has the ability to tailor itself for the new data. ENCVIDC addresses the QUIC protocol traffic from different browsers and has the ability to tailor itself against the webpage-specific videos.Figure 5.17 shows how ENCVIDC varies in performance when contrasted with the existing techniques. It is clear that ENCVIDC is an efficient technique in predicting the encrypted video stream content.

Table 5.2: Comparison of ENCVIDC with Existing Techniques

paper	**Dubin et al. [32]**	**Dubin et al. [34]**	**Schuster et al. [35]**	**Li et al.[36]**	**Dubin et al. [37]**	**Reed et al. [40]**	**ENCVIDC**
AI	ML	ML	DL	DL	ML	ML	Ml
Learning	supervise	supervise	supervise	supervise	unsupervise	supervise	supervise
Technique	SVM,kNN	SVM,kNN	CNN	CNN,MPN	k-means Clustering	kd tree	Random Forest
Features	flow	flow	flow	flow	NLP BPP words	Statistical	Statistical
Content	General	General	Particular	General	General	Particular	General
Dataset	2700	10,000	140	3000	10,000	42,027	3036
Video sites	You Tube	You Tube	YouTube etc.	You Tube	You Tube	Netflix	You Tube etc.
Removal of Audio	✓	✓	✓	✓	✓	✗	✗
Accuracy	98%	95%	(90-98%)	97%	79%	99.5%	98.13%
Error	2%	5%	(2-10%)	3%	21%	0.5%	1.87%
QUIC streaming	✗	✗	✗	✗	✗	✗	✓
Model Updating	✗	✗	✗	✗	✗	✗	✓
Feature Reduction	✗	✗	Greedy search ✓	✗	PCA ✓	✗	✗
Compressed Traffic	✗	✗	✗	✗	✗	✗	✓
Real-time network traffic	✗	✗	✗	✗	✗	✓	✓

Chapter 6

Conclusion and Future Work Track

This chapter is concerned with the future work recommendations for the research describe in the above chapters.

6.1 Conclusions

Over the past couple of years, individuals have seen the phenomenal digital revolutions associated with the internet. As a result, the network traffic is becoming diverse in nature with time. However, such diversity of network traffic and sophisticated Information Technology (IT) innovations are increasing cyber security issues at a greater pace. At present, adversaries are capable of sniffing confidential information from the traffic passively. Therefore, for addressing security-related challenges, different browsers like Google, Firefox and Opera etc. nowadays are implementing Transport Layer protocols (TCP/UDP) with Security Layer protocols (TLS/SSL) for data streaming services, thereby providing a secure communication path among the communicating parties.

The traditional users of the internet are of the view that their online viewed activities are hidden and segregated from the outside world as encryption or similar counter measures tools have been employed over the internet. However, in recent times, with the introduction of encrypted video traffic content classification techniques, the confidential information about the users can be extracted from the encrypted content as well. The existing techniques present in the literature pose some limitations, which affect the per-

formance. These may include inapplicability to real-time network traffic, adherence to the compression nature of the traffic and no new data feedback to the already trained models etc. Therefore, the availability of a robust encrypted video content classification technique is the desire need of today's society due to several factors leading towards the fulfillment of desired and required security related objectives of the internet network.

ENCVIDC proposed in this paper, is an efficient approach for encrypted video traffic content taxonomy that classifies the content of encrypted video traffic flowing over the internet with an accuracy of more than 98% within a second without any direct interaction with the network communicating device.

6.2 Future Recommendations

Nevertheless, some ideas can be introduced for the auxiliary enhancement of our methodology for instance to employ the scheme on cloud, the classifier can be trained on dataset can be encrypted by Homomorphic encryption [44].Through this way, it will be easy to share the classifier to a large amount of organizations without leaking any data.

References

[1] Chen, J., Miao, F., Wang, Q. (2007, April). SSL/TLS-based Secure Tunnel Gateway System Design and Implementation. In 2007 International Workshop on Anti-Counterfeiting, Security and Identification (ASID) (pp. 258-261). IEEE.

[2] Wright, C. V., Monrose, F., Masson, G. M. (2006). On inferring application protocol behaviors in encrypted network traffic. Journal of Machine Learning Research, 7(Dec), 2745-2769.

[3] Draper-Gil, G., Lashkari, A. H., Mamun, M. S. I., Ghorbani, A. A. (2016, February). Characterization of encrypted and vpn traffic using time-related. In Proceedings of the 2nd international conference on information systems security and privacy (ICISSP) (pp. 407-414).

[4] Bar-Yanai, R., Langberg, M., Peleg, D., Roditty, L. (2010, May). Realtime classification for encrypted traffic. In International Symposium on Experimental Algorithms (pp. 373-385). Springer, Berlin, Heidelberg.

[5] Li, W., Moore, A. W. (2007, October). A machine learning approach for efficient traffic classification. In 2007 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (pp. 310-317). IEEE.

[6] Kumar, P. A., Chandramathi, S. (2015). Intelligent video QoE prediction model for error-prone networks. Indian Journal of Science and Technology, 8(16), 1.

[7] Lian, S., Wang, X., Sun, J., Wang, Z. (2004, October). Perceptual cryptography on wavelet-transform encoded videos. In Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004. (pp. 57-60). IEEE.

[8] Lian, S., Sun, J., Wang, Z. (2004, June). Perceptual cryptography on SPIHT com-

pressed images or videos. In 2004 IEEE International Conference on Multimedia and Expo (ICME)(IEEE Cat. No. 04TH8763) (Vol. 3, pp. 2195-2198). IEEE.

[9] Shi, C., Bhargava, B. (1998, October). An efficient MPEG video encryption algorithm. In Proceedings Seventeenth IEEE Symposium on Reliable Distributed Systems (Cat. No. 98CB36281) (pp. 381-386). IEEE.

[10] Stockhammer, T. (2011, February). Dynamic adaptive streaming over HTTP– standards and design principles. In Proceedings of the second annual ACM conference on Multimedia systems (pp. 133-144).

[11] Sodagar, I. (2011). The mpeg-dash standard for multimedia streaming over the internet. IEEE multimedia, 18(4), 62-67.

[12] Witten, I. H., Frank, E. (2002). Data mining: practical machine learning tools and techniques with Java implementations. AcmSigmod Record, 31(1), 76-77.

[13] Li, F., Chung, J. W., Claypool, M. (2018, June). Silhouette: Identifying youtube video flows from encrypted traffic. In Proceedings of the 28th ACM SIGMM Workshop on Network and Operating Systems Support for Digital Audio and Video (pp. 19-24).

[14] Andersson, R. (2017). Classification of video traffic: an evaluation of video traffic classification using random forests and gradient boosted trees.

[15] Liu Y., Li S., Zhang C., Zheng C., Sun Y., Liu Q. (2020) ITP-KNN: Encrypted Video Flow Identification Based on the Intermittent Traffic Pattern of Video and K-Nearest Neighbors Classification. In: Krzhizhanovskaya V. et al. (eds) Computational Science – ICCS 2020. ICCS 2020. Lecture Notes in Computer Science, vol 12138. Springer, Cham

[16] Wassermann, S., Seufert, M., Casas, P., Gang, L., Li, K. (2019, June). Let me decrypt your beauty: Real-time prediction of video resolution and bitrate for encrypted video streaming. In 2019 Network Traffic Measurement and Analysis Conference (TMA) (pp. 199-200). IEEE.

[17] Seufert, M., Wassermann, S., Casas, P. (2019). Considering user behavior in the quality of experience cycle: Towards proactive QoE-aware traffic management. IEEE Communications Letters, 23(7), 1145-1148.

[18] Kumar, P. A., Chandramathi, S. (2015). Intelligent video QoE prediction model for error-prone networks. Indian Journal of Science and Technology, 8(16), 1.

[19] Shi, Y., Ross, A., Biswas, S. (2018). Source identification of encrypted video traffic in the presence of heterogeneous network traffic. Computer Communications, 129, 101-110.

[20] Michie, D., Spiegelhalter, D. J., Taylor, C. C. (1994). Machine learning. Neural and Statistical Classification, 13(1994), 1-298.

[21] Alpaydin, E. (2020). Introduction to machine learning. MIT press.

[22] Retrieved from:https://en.wikipedia.org/wiki/Machine-learning

[23] Kotsiantis, S. B., Zaharakis, I., Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 160(1), 3-24.

[24] Liaw, A., Wiener, M. (2002). Classification and regression by randomForest. R news, 2(3), 18-22.

[25] Witten, I. H., Frank, E. (2002). Data mining: practical machine learning tools and techniques with Java implementations. AcmSigmod Record, 31(1), 76-77.

[26] Prasad, A. M., Iverson, L. R., Liaw, A. (2006). Newer classification and regression tree techniques: bagging and random forests for ecological prediction. Ecosystems, 9(2), 181-199.

[27] Freund, Y., Schapire, R. E. (1996, July). Experiments with a new boosting algorithm. In icml (Vol. 96, pp. 148-156).

[28] Zhu, X., Goldberg, A. B. (2009). Introduction to semi-supervised learning. Synthesis lectures on artificial intelligence and machine learning, 3(1), 1-130.

[29] Papadamou, K., Papasavva, A., Zannettou, S., Blackburn, J., Kourtellis, N., Leontiadis, I., ...Sirivianos, M. (2019). Disturbed youtube for kids: Characterizing and detecting disturbing content on youtube. arXiv preprint arXiv:1901.07046.

[30] Agrawal, S., Sureka, A. (2013, December). Copyright infringement detection of music videos on youtube by mining video and uploader meta-data. In International Conference on Big Data Analytics (pp. 48-67). Springer, Cham

[31] Kandakatla, R. (2016). Identifying offensive videos on YouTube.

[32] Dubin, R., Dvir, A., Hadar, O., Pele, O. (2016). I know what you saw last minute-the chrome browser case. Black Hat Europe.

[33] Noble, W. S. (2006). What is a support vector machine?.Nature biotechnology, 24(12), 1565-1567

[34] Dubin, R., Dvir, A., Pele, O., Hadar, O. (2017). I know what you saw last minute—encrypted http adaptive video streaming title classification. IEEE transactions on information forensics and security, 12(12), 3039-3049.

[35] Schuster, R., Shmatikov, V., Tromer, E. (2017). Beauty and the burst: Remote identification of encrypted video streams. In 26th USENIX Security Symposium (USENIX Security 17) (pp. 1357-1374)

[36] Li, Y., Huang, Y., Xu, R., Seneviratne, S., Thilakarathna, K., Cheng, A., ...Jourjon, G. (2018, November). Deep Content: Unveiling video streaming content from encrypted WiFi traffic. In 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA) (pp. 1-8). IEEE.

[37] Dvir, A., Marnerides, A. K., Dubin, R., Golan, N. (2019, February). Clustering the Unknown-The YouTube Case. In 2019 International Conference on Computing, Networking and Communications (ICNC) (pp. 402-407). IEEE.

[38] Ding, C., He, X. (2004, July). K-means clustering via principal component analysis. In Proceedings of the twenty-first international conference on Machine learning (p. 29).

[39] Nadkarni, P. M., Ohno-Machado, L., Chapman, W. W. (2011). Natural language processing: an introduction. Journal of the American Medical Informatics Association, 18(5), 544-551.

[40] Reed, A., Kranch, M. (2017, March). Identifying HTTPS-protected Netflix videos in real-time. In Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy (pp. 361-368).

[41] Lin, J. (1991). Divergence measures based on the Shannon entropy. IEEE Transactions on Information theory, 37(1), 145-151.

[42] Blum, A. L., Langley, P. (1997). Selection of relevant features and examples in machine learning. Artificial intelligence, 97(1-2), 245-271.

[43] Hesterman, J. Y., Caucci, L., Kupinski, M. A., Barrett, H. H., Furenlid, L. R. (2010). Maximum-likelihood estimation with a contracting-grid search algorithm. IEEE transactions on nuclear science, 57(3), 1077-1084.

[44] Tebaa, M., El Hajji, S., El Ghazi, A. (2012, April). Homomorphic encryption method applied to Cloud Computing. In 2012 National Days of Network Security and Systems (pp. 86-89). IEEE.

[45] Rahimi, A., Recht, B. (2008). Random features for large-scale kernel machines. In Advances in neural information processing systems (pp. 1177-1184).

[46] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.

[47] Zhang, Z. (2016). Introduction to machine learning: k-nearest neighbors. Annals of translational medicine, 4(11).

[48] Langley, A., Riddoch, A., Wilk, A., Vicente, A., Krasic, C., Zhang, D., ... Bailey, J. (2017, August). The quic transport protocol: Design and internet-scale deployment. In Proceedings of the Conference of the ACM Special Interest Group on Data Communication (pp. 183-196).

[49] Carlucci, G., De Cicco, L., Mascolo, S. (2015, April). HTTP over UDP: an Experimental Investigation of QUIC. In Proceedings of the 30th Annual ACM Symposium on Applied Computing (pp. 609-614).

[50] Benettin, G., Galgani, L., Strelcyn, J. M. (1976). Kolmogorov entropy and numerical experiments. Physical Review A, 14(6), 2338.

[51] Elkeelany, O., Matalgah, M. M., Sheikh, K. P., Thaker, M., Chaudhry, G., Medhi, D., Qaddour, J. (2002, April). Performance analysis of IPSec protocol: encryption and authentication. In 2002 IEEE International Conference on Communications. Conference Proceedings. ICC 2002 (Cat. No. 02CH37333) (Vol. 2, pp. 1164-1168). IEEE.

[52] Kumar, S., Turner, J., Williams, J. (2006, December). Advanced algorithms for fast and scalable deep packet inspection. In 2006 Symposium on Architecture For Networking And Communications Systems (pp. 81-92). IEEE.

[53] Barlow, H. B. (1989). Unsupervised learning. Neural computation, 1(3), 295-311.

[54] Sutton, R. S., Barto, A. G. (2011). Reinforcement learning: An introduction.

[55] Hackeling, Gavin. Mastering Machine Learning with scikit-learn. Packt Publishing Ltd, 2017.

[56] Varoquaux, G., Buitinck, L., Louppe, G., Grisel, O., Pedregosa, F., Mueller, A. (2015). Scikit-learn: Machine learning without learning the machinery. GetMobile: Mobile Computing and Communications, 19(1), 29-33.

[57] [57] Gu, J., Wang, J., Yu, Z., Shen, K. (2018, April). Walls have ears: Traffic-based side-channel attack in video streaming. In IEEE INFOCOM 2018-IEEE Conference on Computer Communications (pp. 1538-1546). IEEE.