Trust-Based Video Management Framework for Social Multimedia Networks

Badr Eddine Mada, Miloud Bagaa, and Tarik Taleb

***Abstract*—Social multimedia networks (SMNs) have attracted much attention from both academia and industry due to their impact on our daily lives. The requirements of SMN users are increasing along with time, which make the satisfaction of those requirements a very challenging process. One important challenge facing SMNs consists of their internal users that can upload and manipulate insecure, untrusted, and unauthorized contents. For this purpose, controlling and verifying content delivered to end users is becoming a highly challenging process. So far, many researchers have investigated the possibilities of implementing a trustworthy SMN. In this vein, the aim of this paper is to propose a framework that allows collaboration between humans and machines to ensure secure delivery of trusted video content over SMNs while ensuring an optimal deployment cost in the form of CPU, RAM, and storage. The key concepts beneath the proposed framework consist in assigning to each user a level of trust based on his/her history, creating an intelligent agent that decides which content can be automatically published on the network and which content should be reviewed or rejected, and checking the videos' integrity and delivery during the streaming process. Accordingly, we ensure that the trust level of the SMNs increases. Simultaneously, efficient capital expenditure and operational expenditures can be achieved.**

***Index Terms*—Social multimedia network, video streaming, trust model, and trust management.**

I. Introduction

THE recent advances in the Internet have resulted in the emergence of many web applications and social multimedia networks (SMN). These applications (e.g., Facebook, Twitter, and Google) have revolutionized the use of the Internet as a tool to interconnect people over the world. The features implemented by these service providers have been making communication between people easier. Service providers have granted to the users more flexibility for interacting among themselves and exchanging different social information.

Thanks to these services, users can easily discuss their ideas and opinions remotely, publish new articles, and meet new persons. Moreover, they have allowed business and organizations to advertise for their products over the world and to directly contact their customers. In addition to these social networks, other web applications, such as Youtube, Dailymotion, and Vimeo, have enabled the exchange of different contents, including text, images, and videos among different entities connected to their services. The evolution of the Internet and distributed systems has led researchers to implement applications that serve video on demand (VOD) on top of the peer-to-peer (P2P) networks [1]–[3].

VOD and videos live streaming systems are gaining momentum in SMN. They have enabled the appearance of many multimedia-centric services such as video conferencing applications, online meeting applications, massive open online courses (MOOC) as well as other use cases in e-health and e-teaching [4]. Such services attract and connect millions of users worldwide. The providers of these services have enabled countless features that allow users to interact among themselves by creating and sharing different contents (e.g, videos, text, and images).

However, by allowing this, the nodes composing the social networks, users and machines, generate a huge amount of data, which can be uncontrolled, unsecured and untrusted [5], [6]. Such amount of generated data are causing a congestion to the networks [7], [8] and posing a new security challenge to the service providers: it becomes hard to handle and analyze all content traversing their networks. To tackle this problem, many research efforts have been conducted so far for mitigating the upload of malicious data to SMNs. Diverse data analytics applications have been proposed and developed with the goal to create a trustworthy SMN [9], [10].

The researchers' vision of trustworthy SMNs [11] lies in achieving certainty, authenticity, and security of data exchanged throughout social network nodes [12], [13]. In this vein, many trust models and reputation systems have emerged [14]–[16] with the goal to limit the spread of unsecured data. Generally, trust models and reputation systems are designed to assign a score to each entity in the network and establish trust among them. This score may help users to make a proper decision on buying an item from an online store, selecting a service provider or recommending a service to other users. Additionally, the trust

Manuscript received July 23, 2018; revised October 25, 2018; accepted December 3, 2018. Date of publication January 16, 2019; date of current version February 21, 2019. This work was supported in part by the ANASTACIA Project, which has received funding from the European Unions Horizon 2020 Research and Innovation Programme under Grant Agreement 731558 and from the Swiss State Secretariat for Education, Research and Innovation, and in part by the Academy of Finland 6Genesis Flagship under Grant 318927. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Zhou Su. *(Corresponding author: Miloud Bagaa.)*

B. E. Mada and M. Bagaa are with the Department of Communications and Networking, School of Electrical Engineering, Aalto University, Espoo 02150, Finland (e-mail: badr.mada@aalto.fi; miloud.bagaa@aalto.fi).

T. Taleb is with the Department of Communications and Networking, School of Electrical Engineering, Aalto University, Espoo 02150, Finland, with the Centre for Wireless Communications, University of Oulu, FI-90014 Oulu, Finland, and also with the Computer and Information Security Department, Sejong University, 143-747 (05006) Seoul, South Korea (e-mail: tarik.taleb@aalto.fi).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2019.2893548

score provides decisional systems with the needed information to execute adequate actions, such as the implementation of certain policies that restraint an entity from using some resources or accessing some services.

The main features that should be taken into consideration while defining a trust model are as follows:

- User history: The only way to predict user behavior is to study and analyze all generated content by different users during their interactions in the network [17]. The user history records may contain relations and links between data [18], these links are valuable for the data analytics applications in order to offer a good user experience.
- Trust calculation: A user's level of trust is one of the important metrics that should be taken into consideration when analyzing users' data. The computation of this value includes the selection of various parameters that characterize the manipulated data [19]. For this reason, there is a need to suggest a realistic model that can capture the characteristics of uploaded data based on the historical behavior of users.
- Users collaboration: Based on the observation that human intelligence is one of the main keys to effectively detect and remove untrusted data, many algorithms and applications have been recently devised for detecting and measuring users' collaborations rate [20]. These algorithms and applications allow users to rate different social multimedia items. Then, the system is able to collect these feedbacks, applies some filtering methods and executes different needed actions.
- Secure content delivery: In a trustworthy social network, every bit of data should be under control. In other words, starting from any node in the network (e.g, user, mobile, or server), the path that the data take to arrive at another node should be secured [21], [22].

Ensuring a secure delivery of trusted videos and preventing users of social networks from manipulating insecure, untrusted and unauthorized contents is a challenging process that needs a high amount of computational power. Currently, the well-known social media networks rely on their users to report unauthorized contents in order to take the different countermeasures. To the best of the authors' knowledge, there is no automatic way to prevent users from uploading insecure, untrusted and unauthorized contents. In this paper, we fill this gap by proposing a generic framework that creates a trustworthy SMN. The main goal of the generic framework is to create a system that is able to provide secure delivery of trusted videos content over social networks with low resources consumption in terms of CPU, RAM, and storage. Indeed, the proposed system reduces the resource utilization, and accordingly the cost, by analyzing only the video content that really needs to be analyzed. The proposed framework explores both the user history and users' collaboration for taking the decision to either make the analytical analysis or not. The framework contains a module that is responsible for calculating the level of trust of each user in the network. Besides the user trust calculation module, the generic framework has: $i)$ a voting service to allow users rewarding trusted clients and penalizing malicious users; $ii)$ an incentive module to remunerate the users for their collaboration; $iii)$ secure videos module that ensures secure delivery of videos; and $iv)$ a video integrity checker service to assure the integrity and timestamping of uploaded videos.

Moreover, an adaptation on the video player, at the client side, is also proposed to take into consideration the new features suggested in the new framework. The update consists of implementing a new functionality at the video player that enables it to communicate with the video integrity checker and verify that the chunks buffered were not altered during the streaming process. Furthermore, the proposed generic framework has a video uploading decision process module that enables checking the quality of the uploaded videos before either accepting the publication or not. Besides the use of historical behavior of users, this module explores two techniques for checking the quality of the uploaded contents: $i)$ analytical checking of the uploaded videos; $ii)$ review checking of the uploaded contents by a set of trusted users. Based on the observation that those techniques are expensive, this module explores the historical behavior of the users with the goal to take the decisions without involving those two techniques. Also, it uses infinite Discrete Markov Decision Process (DMDP) for taking the decisions to either publish or not an uploaded video. Thanks to DMDP, the module is able to decide to either analytically check the contents or send them to an external reviewer before publishing or deny the publication of the uploaded contents.

The remaining of this paper is structured as follows. Section II discusses the previous works proposed in the literature. Section III describes the proposed framework. Section IV introduces the video uploading decision process module, while section V presents and discusses the simulation results. Finally, the paper concludes in Section VI.

II. Related Work

In this section, we briefly present the research works that are most relevant to our proposed framework including trustworthiness and social interactions of SMNs. Most research work, published concerning the trustworthiness among entities in a network, have studied the trust level in a way that they compute the degree of trust amongst users or nodes composing the network [19], [23]–[26]. In this paper, the trust is defined as how much the system trusts each user, and how this level of trust affects the total cost spent for different resources in order to filter and analyze the uploaded data. In [23], authors have discussed the trust and reputation system (TRS) in e-Health. They characterize the trust as not bidirectional between entities; trust is subject to the expectations and is partially transitive. Moreover, the authors presented some possible attacks on the trust model, in particular, $i)$ the bad mouthing attack that occurs when an untrusted entity tries to hurt the reputation of another entity; and $ii)$ the collusion attack that emerges when a group of entities tries to boost each other's reputation.

A Machine learning (ML) based approach is used in [19] to calculate the trust score for the different nodes of the social network. The logistic regression is used to train the neural network. The main reason beneath using such a model is the flexibility of ML solutions that can be adapted to different networks and platforms. The authors also introduced a method to effectively select the features that describe the data. In the same way, the authors in [27] used ML-based algorithms to mine the trust and distrust relationships in a social web application. In order to train their model to do some predictions, they introduced four inputs factors. The first factor, named Knowledge-based trust, combines the number of satisfactions between two given nodes. The second factor, named similarity-based trust, shows the degree of similarity between truster and trustee. The third factor, named reputation-based trust, represents the social importance of an entity in the network. Finally, the fourth factor, dubbed personality-based trust factor, shows a user's tendency to trust another user.

Authors in [24] proposed a method based on user cosine similarity [28] in order to calculate the trust value. This calculated value can be used to filter the neighbors and predict a recommendation items to another similar user. In their model, the authors considered that the trust value is transitive and can be transferred from a user to another. Wang *et al.* in [25] proposed a trust model based on a Bayesian trust algorithm for self-organizing networks. The main idea behind this method is counting the number of successful and unsuccessful messages. In this work, the authors presented the trust as a tree dimensions vector. The first dimension of the vector is the connectivity, which is the capability of a node to connect another node in the network. The second dimension is fitness. It describes the behavior of a node and can help in detecting malicious nodes. The last dimension is the satisfaction, this parameter shows how much a node is satisfied by the intermediate nodes. By computing the parameters of this vector, each node can calculate the vector trust of other nodes and decide to accept or reject a recommendation from them.

Last but not least, authors in [26] exploited the graph theory to compute the trust and distrust in a network. Their work was inspired by the computation of path probability in random graphs [29]. The graph's edges represent the probability that a path exists between user A and user B. On the other hand, the distrust was inspired by spring embedding graph layout algorithms. The combination of these two algorithms allows the proposed trust model to pull trusted nodes and regroup them in a form of trusted cluster, conversely, untrusted nodes are pushed away.

III. Social Multimedia Network Generic Framework

In this section, we will describe the generic framework proposed in this paper. As depicted in Fig. 1, the proposed framework mainly consists of five modules, which are: i) social network module (SNM); ii) secure video manager module (SVM); iii) video integrity checker module (VICM); iv) video uploading decision process (VUDP) module, and finally v) incentive module (IM). Those five modules work in a unified manner for offering a secure user-friendly system that gives to the social multimedia network users the flexibility for managing different videos in an efficient and secure manner. The management of a video includes its upload, storage, sharing, as well as streaming processes. Fig. 2 depicts a sequence diagram that shows the interactions between the SMN users and the different components of the framework, as well as the interactions among the components in order to serve the users' requests.

First of all, a user should be authenticated to the system using the social network module. To access the system, the user through the user interface sends an authentication message to the SNM (Fig. 2: arrow 1). At the moment when the user gets the authorization, he/she can perform different social interactions with other authenticated users including chats, messaging, etc. The IM is responsible to compute the incentives for different users by taking into account their interactions with other users and their collaborations to detect malicious users and untrusted content. One of the important interaction with the system is the upload and the secure delivery of stored videos. Moreover, according to the user interaction, the proposed framework, more precisely the user trust calculation sub-module (i.e., that belongs to the VUDP module) will update the trust score of that user. When an authenticated user plans to upload a new video, the user interface, on behalf of that user, requests a token from the SNM. The received token will be used for ensuring secure authentication through the SVM module (Fig. 2: arrow 5). SVM checks the validity of the received token by consulting SNM (Fig. 2: arrow 6). If the received token is valid, SVM will proceed with the upload of the video. Otherwise, a failure message will be generated and forwarded to the end user.

The proposed framework is designed in a way to increase the average trust score of the uploaded videos. For this purpose, the proposed framework aims to allow only trusted videos to be uploaded to the system as much as possible. During the video upload process, the SVM needs to consult VUDP module, more precisely the decision algorithm sub-module, before authorizing the user to upload any video (Fig. 2: arrow 13). The decision process runs the decision algorithm that should make the decisions after checking the trust score of that user and his videos. The analysis of a video needs different techniques including video processing, such as object detection [30]–[32], and machine learning techniques [33]. Mainly, those techniques require a lot of processing power and a long execution time [34]. Based on the observation that the analysis of a video is consuming a lot of resources, it is worthless to analyze every video uploaded to the system. For this reason, the smart algorithm sub-module should use an efficient technique for analyzing videos only if needed. The smart algorithm sub-module mainly explores the user trust information received from the user trust calculation module, for taking the decision to either perform the analytical video checking or not. Moreover, the smart algorithm sub-module can also contact a set of trusted users in order to perform manual checking of the uploaded videos if needed. The manual checking of the uploaded videos includes both the already analytically-checked videos and the not-checked ones. Then, according to the response received from the video uploading decision process module, SVM decides to either accept the video upload from that particular user or not (Fig. 2: arrow 13).

In the case that the SVM accepts the upload of the video, the user interface sends the target video chunk by chunk to SVM. This will enable the resumable upload of that video. For

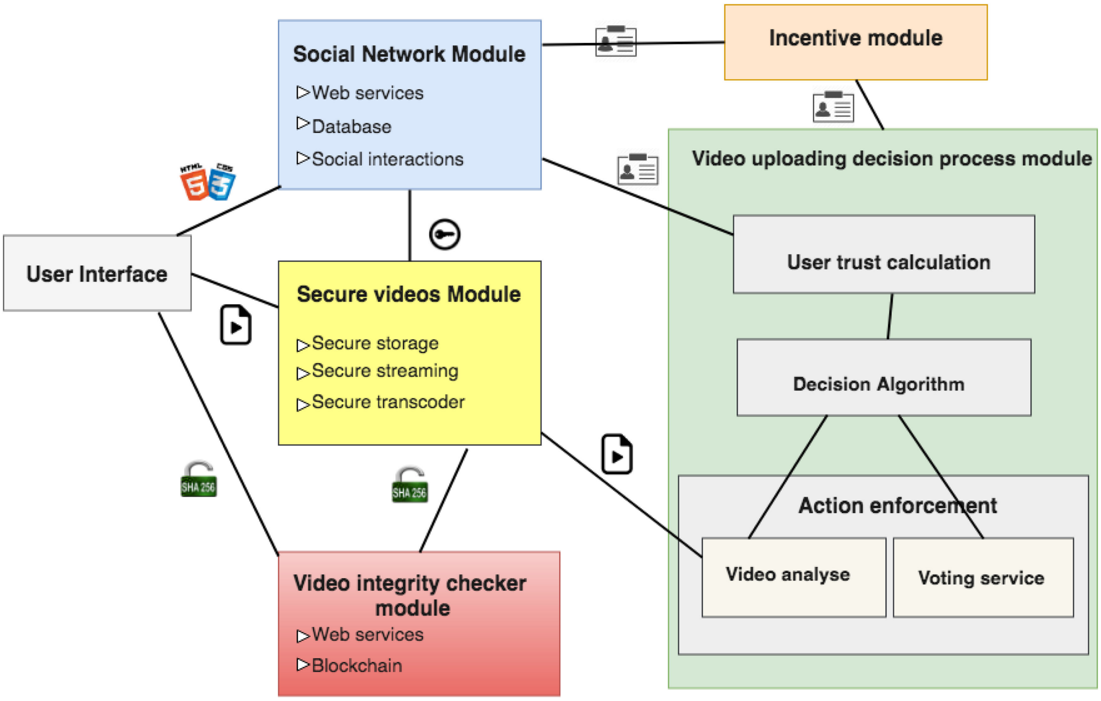


Fig. 1. Main overview of the architecture of a trustworthy SMN.

instance, for any reason, if the connection drops between the user interface and SVM, as long as the token is still valid, the user can upload only the remaining chunks of that video when the connection is reestablished instead of uploading the whole video. When SVM receives the chunks, it encodes them to different qualities. It then stores them in secure storage. Moreover, for each transcoded video chunk, the secure video module computes its hash and sends it to the video integrity checker module (VICM) for further use. VICM saves the received chunks in a BLOCKCHAIN, such as originstamp.org API or a private BLOCKCHAIN system, for ensuring the integrity of the streaming process later. This strategy helps the proposed system for ensuring that the signature and the timestamps of each chunk are stored in a shared database accessible by different actors.

When a user wants to watch that specified video, he first needs to be authenticated with SNM (Fig. 2: box User authentication). After successful authentication, the user interface requests a token that should be forwarded later to SVM. Similar to the previous case, SVM checks the token by consulting SNM. After the successful authentication of that user, the video player in the user interface starts requesting the chunks one by one from SVM. In order to check the integrity of different chunks, the user interface could compute the hash of the chunks and compare them to the one already stored in the shared database. In order to mitigate the overhead on the user interface, a smart strategy should be also implemented at the client side (i.e., user interface), enabling to check the integrity only of a small number of chunks. For this reason, the smart selection algorithm, at the user interface, selects a list of chunks that require the integrity check. During the streaming process, when the user interface receives a chunk, which is in that list, a request should be sent to VICM. After receiving the hash of that chunk from VICM, the user interface computes and compares the hash of that chunk to the received one. If both chunks have the same hash value, the chunk will be streamed to the user. Otherwise, an alert will be generated and forwarded to the user and SNM. After receiving the alert, SNM will update the trust of the video owner through the trust calculation module. Other measurements could be also applied.

While the remaining of this section summarizes the objectives and functionalities of SNM, SVM, VICM, and IM modules, Section IV describes the VUPD module in a detailed manner.

A. *Social Network Module (SNM)*

This module is the first component that interacts with the users. It permits them to do all kind of social interactions, such as the upload of videos, the post of comments, and the sharing of different videos. This module is composed of many micro-services that communicate with each other in order to offer a user-friendly application that fulfills the end-users needs. The main micro-services are i) the web server that responds to the users' requests, ii) a database that stores all information of users and their generated content, iii) a caching micro-service for reducing the response time and allowing the users to have good experiences while interacting with the system, iv) a message broker that allows the communication between the different components, and v) a central authentication service that authenticates the users and gives them the right to request other services.

B. *Secure Videos Module (SVM)*

This module allows authorized users to upload their media files to the secure storage, as well as it allows the social network

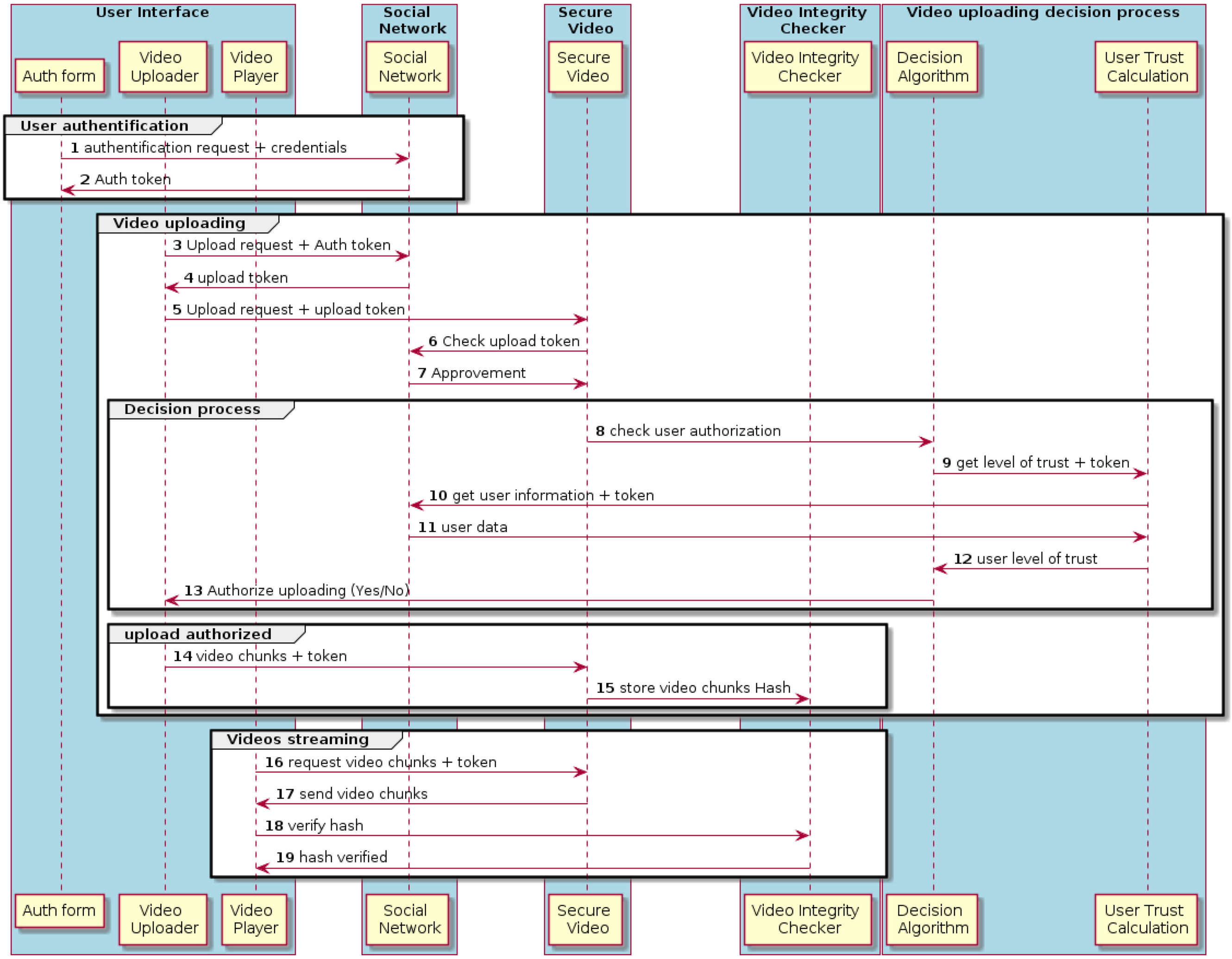


Fig. 2. Sequence diagram for a secure upload of videos.

users to watch the videos streamed on demand from the secure streaming. The SVM consists of three components:

- Secure storage: this component mainly works as follows: first of all, an authorized user sends an upload request to the SNM. Then, the social network module (SNM), more precisely the central authentication micro-service, generates and stores a unique token in the database, and then sends it to the user as a response. The user starts sending the video chunks to the storage server while including that token within the messages sent. The storage server (SS) checks the received token and then decides either to accept or reject the upload.

  This component adopts the HTTP live stream (HLS) for serving diverse users with different resolutions adapted to their network bandwidth and devices. Also, this component uses the Rivest Cipher 4 (RC4) algorithm in order to encrypt the video chunks sent to the end users.
- Secure transcoder: this component allows the transcoding of the uploaded videos to different resolutions using softwares such as FFMPEG. Each resolution is subdivided into small chunks of fixed time duration [35]. After the transcoding operation ends, the secure transcoder creates a hash for each chunk and sends that hash to the video integrity checker module (VICM). The VICM saves that hash in a public or private BLOCKCHAIN service as a transaction. The hashed values will be used by the user video player to verify that the chunks received were approved by the system and the chunks were not modified from the time that a user uploaded the video to the secure storage.

C. *Video Integrity Checker Module (VICM)*

The main feature of this module is to allow the timestamping of the chunks generated from an uploaded video. This helps in checking the integrity of these chunks in the future. Formally, the VICM module saves the video content, its signature and its date-time of creation in a trusted and a shared database. Also, this module checks that the file has not been altered or modified thanks to Blockchain technologies. Moreover, the service will be also used from a client (e.g, browser, tablet, smartphone, etc) to verify that the video chunks received were not altered during the streaming process.

D. Incentive Module (IM)

In order to motivate users to review some uploaded videos and decide to publish them or not, an incentive component was created to reward the users for their contributions. This component is responsible for remunerating the reviewers when they make a true vote. A vote is considered true when the decision made at the proposed framework is to publish the uploaded video.

IV. Video Uploading Decision Process

In this section, we describe the video uploading decision process (VUDP) module, depicted in Fig. 1. The main responsibility of this module is the efficient control of the trustworthiness of the uploaded videos while minimizing the computation overhead. Formally, the VUDP module should increase the average level of trust for all uploaded videos with minimum efforts. To measure the trust level of a video, mainly two methods could be applied. The first method explores a set of trusted users for reviewing the content of that video and then sending back their feedbacks about the content. In this case, the VUDP module aggregates the received report before making the final decision about the content. The second method analytically evaluates the content of that video using different mathematical techniques and software. For instance, the trust level of an uploaded video could be calculated by extracting the text from the video and applying suitable object recognition techniques. The proposed framework can use both methods or only one of them before making any decision about the trustworthiness of a video. Both methods could be even skipped if the user has a high trust value in order to save the efforts and incurred costs. As aforementioned in Section III, VUDP consists of four sub-modules: i) user trust calculation; ii) decision algorithm; iii) video analysis; and iv) voting service. We describe these sub-modules hereunder.

A. Trust Calculation Module

This sub-module has the responsibility to compute the trustworthiness of different users. For this reason, it keeps monitoring the behavior of each user by taking into consideration his/her social interactions with other users. These social interactions include, but not limited to, the following parameters: i) the number of followers (NOF); ii) the number of true votes (NOTV) received from trusted users through the voting service sub-module; iii) the percentage of true reports (PTR) received from different users of the social network; iv) the percentage of likes (POL) received from the user network mainly his friends; and v) the average trust of published videos (ATPV). For the sake of simplicity, the trust value of each user is computed using a weighted sum function of the different parameters. However, any more sophisticated method can be also used with slight modification. For instance, the entropy of Shanon can be also applied to these parameters for computing the trust degree of each user. In what follows, we will show how the trust values of users and videos are computed.

Let $\mathcal{U}$ denote the set of users composing a social network and $\mathcal{X}$ the vector that reflects the scores of the social interaction parameters of different users (e.g., NOF, NOTV, PTR, POL and ATPV). The values of $\mathcal{X}$ should be defined according to the importance of each parameter. Let $\mathcal{L}$ denote the size of the vector $\mathcal{X}$. Let $\mathcal{X}_i$, for $i \in [1, \mathcal{L}]$, denote the i^{th} of vector $\mathcal{X}$. Formally, the score of a user can be defined as follows:

$$\mathbf{TL}_{user}(X) = \frac{\sum_{i=1}^{\mathcal{L}} \omega_i x_i}{\sum_{i=1}^{\mathcal{L}} \omega_i} \tag{1}$$

where ω_i denotes the weigh (i.e., importance) of the i^{th} element of the vector $\mathcal{X}$.

From another side, the trust level of an uploaded video would be computed by considering different objects and texts included in that video. For this purpose, different text and object recognition tools would be applied. The detected objects and texts can be compared to a predetermined list of malicious objects and texts. Formally, for each detected object or text, by exploring the Entropy of Shanon, the probability of a content to be malicious is computed (i.e., belong to the malicious list). The level of trust of a given video could be calculated using the naive Bayes classifier as:

$$\mathbf{TL}_{video_i} = \mathbf{P}(T/W = \{w_i : i \in N\}) = \mathbf{P}(T) \times \prod_{i=1}^{N} \mathbf{P}(w_i/T) \tag{2}$$

where

- **W** $= \{w_i$: i ∈ N$\}$ is a set containing the extracted words and the objects detected.
- **P**(T) is the probability that any word or object is not malicious.
- **P**(w_i/T) is the probability that the word or the object w_i belongs to a trusted class.
- **N = Count(W)** denotes the number of words and objects extracted from the considered video.

B. Voting System Module

The voting service is one of the main components of the system. It allows users to review and vote certain videos in order to be published or not. It also permits users to re-establish their trust level. The set of reviewers is selected according to a method that ensures that there is always a sufficient number of reviewers. The method also allows a subset of users with low values of trust to re-establish their reputations and gradually increase their factor of trust. Moreover, the voting service collects votes and sends the gathered data to the decision making algorithm. The decision algorithm explores the received feedbacks to take decisions on whether to publish or not a video.

C. Decision Algorithm

1) Description: The massive data, mainly videos, shared on SMNs by untrusted users engender a huge amount of resources consumed in terms of CPU, RAM, and storage. Moreover, uploading and manipulating the insecure, untrusted and unauthorized contents by the network nodes could have a negative impact on the whole social multimedia networks. Thus, there is a need to control and verify all contents uploaded to SMNs. However, the verification process could consume even more resources in terms of CPU and memory, which could dramatically affect the CAPEX and OPEX of SMNs. Decision Algorithm is proposed, herein, in order to mitigate the overhead of the verification process while ensuring that the uploaded contents have a high trust value. Basically, the resource utilization is related to the level of trust of each user and the average trust of the whole social multimedia network. If the level of trust is high on the network, then the resources needed to filter the data are low and vice versa. The level of trust assigned to each user is described in Section IV-A.

Based on the observation that checking, controlling and analyzing the uploaded contents are very expensive to process, there is a need to define a smart strategy to analyze only a subset of uploaded videos. The choice of policy to apply whether to highly analyze and store the uploaded video or not depends mainly on the level of trust of each user and the average trust of the whole network. The decision of choosing the optimal policy influences the total cost and the average trust of the system. In what follows, we define the estimated cost of the CPU utilization for analyzing the contents of a video i:

$$Cost_{CPU}(video_i) = C_{CPU} \times t_{CPU_i} \tag{3}$$

where

- t_{CPU_i} is the time required to analyze Video i.
- C_{CPU} is the cost of using the CPU for one unity of time.

Meanwhile, the estimated RAM utilization cost of a video i is computed as follows:

$$Cost_{RAM}(video_i) = C_{RAM_i} \times t_{RAM_i} \tag{4}$$

where

- t_{RAM_i} denotes the time required to analyze a given video i.
- C_{RAM_i} denotes the cost of using the RAM for processing the video i.

For each policy $(\mathcal{P})$ selected by the system, there is a total expected cost defined by:

$$TC^{\mathcal{P}} = \sum_{i=1}^{\infty}(Cost_{CPU}(video_i) + Cost_{RAM}(video_i)) \tag{5}$$

As the aim of the decision algorithm is to find the optimal policy for increasing the average level of trust of the uploaded videos while reducing the total expected cost, the average trust level of the uploaded videos can be defined as follows:

$$ATL_{videos} = \frac{\sum_{i \in V} LT_{video_i}}{|V|} \tag{6}$$

where V is the set of all uploaded videos. Meanwhile, the minimal expected value of the total cost is defined as:

$$TC^{*} = inf_{\mathcal{P}} TC^{\mathcal{P}} \tag{7}$$

The Decision Algorithm should take the right decisions for increasing the average trust values of uploaded videos ATL_{videos} while reducing the expected total cost of $TC^{\mathcal{P}}$. Those decisions can vary from analytically checking the content of the videos till asking the assistance from a set of trusted reviewers. A decision can even accept the upload of a video without analytical checking and/or manual reviewing if the video owner has a higher trust value than ATL_{videos}.

2) Model Formulation and Decisions Making: As aforementioned, the average trust level of a social multimedia network influences the total expected cost for keeping a satisfactory trust level of the system. Basically, the higher the average trust level of a social multimedia network is, the lower the incurred cost becomes. In this sub-section, our focus is on how to design the upload model for increasing the network trust with minimal cost. The Decision Algorithm decides either to publish an uploaded video or not according to the user and the network trust level. Moreover, it can decide to either analyze the content of the uploaded videos or send them to a subset of trusted reviewers for getting their feedbacks about these videos. Basically, the Decision Algorithm is designed in a way to work for a while, and the number of the users' requests is undefined and unlimited. For this reason, in order to achieve optimal decision policies, the Decision Algorithm employs infinite horizon Discrete Markov Decision Process (DMDP) [36]. The latter is designed to evaluate infinite sequences of rewards at all states. The generated policy from DMDP will help the Decision Algorithm for getting a specified action at each situation or state.

The proposed model explores the received information from the user trust calculation sub-module to generate an optimal policy for an uploaded video from a specified user. The framework keeps monitoring the behavior of different users and then updates the user trust calculation sub-module about different social transactions. Note that when a set of reviewers or normal users send a negative opinion about a specified user, the trust level of that user can be affected. Basically, the transition probabilities and rewards of the DMDP are adjusted according to the trust level of that user. The higher the trust level of that user is, the more likely to publish his videos without analytical and/or manual checking. The proposed model is able to impact the average network trust by publishing the videos with high trust values and preventing the insecure, untrusted and unauthorized contents to be uploaded. Moreover, the proposed model should reduce both false positive and false negative when making the decisions. In other words, the system should prevent the user from: i) publishing insecure, untrusted or unauthorized contents; or ii) not publishing contents with high trust values.

Fig. 3 depicts the DMDP used at the Decision Algorithm sub-module for making decisions about each uploaded video. Let $\mathcal{S}_t$ describe the evaluation of the system state and $\mathcal{S}$ denote the state space. We denote by $\mathcal{A} = \{\text{UPL}, \text{ANC}, \text{SANC}, \text{PUB}, \text{NPUB}\}$ the set of actions used for making the decisions on either to publish or not the video uploaded from a specified user.

The proposed framework is a closed loop control system, whereby the decisions that are taken for publishing or not a specified content will have an impact on the trust level of the user and the network. The latter will have a considerable impact on the rewards and transition probabilities of different actions in the DMDP. The action UPL refers to the upload of a new upcoming video, whereas the action ANC refers to analytical

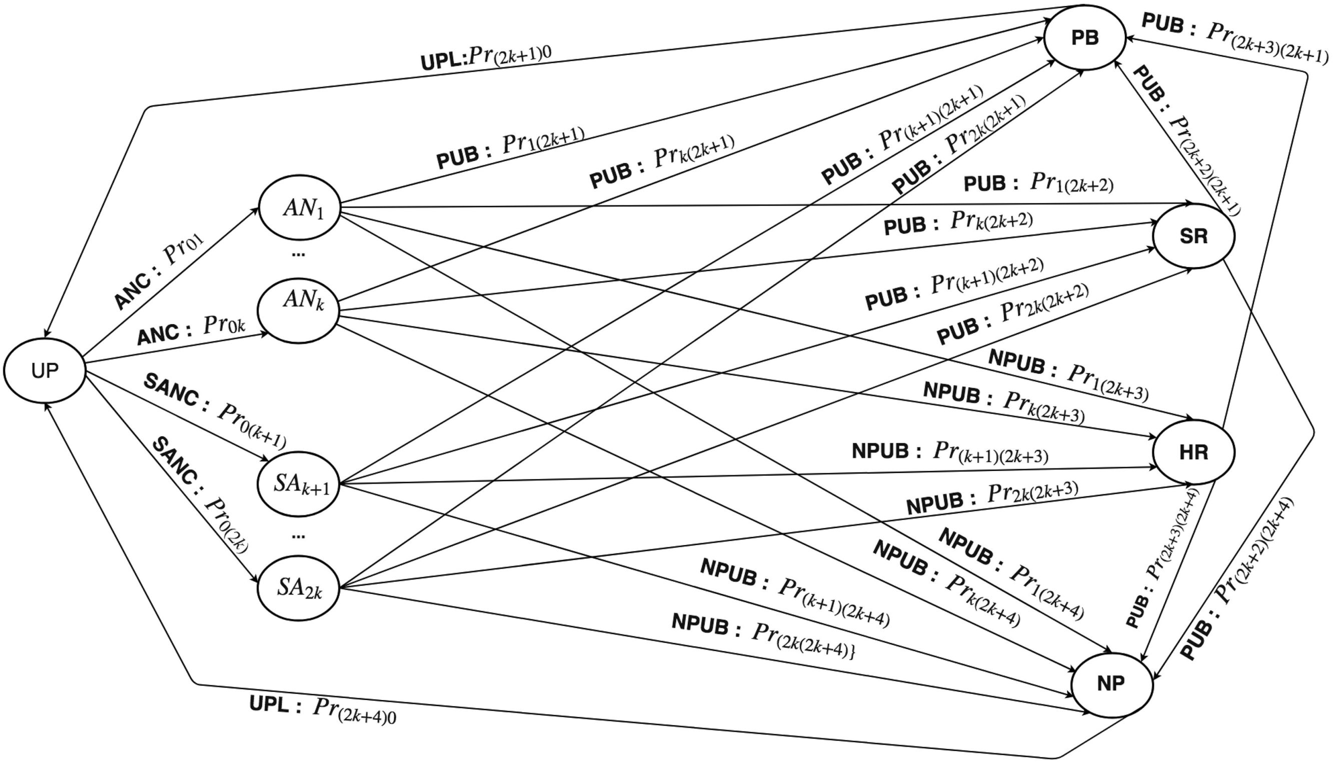


Fig. 3. Infinite horizon Discrete Markov Decision Process employed at the Decision Algorithm.

checking of the uploaded video and the action SANC refers to skipping the analytical checking process. The action PUB refers to publishing the video in the system. However, before publishing the video, it can be reviewed by a subset of reviewers. The probability to either directly publish a video or publish it after a review process varies from a user to another according to his trust level. Finally, the action NPUB refers to not publish the content in the system. Content could be not published directly or after receiving negative feedback from the reviewers. Similar to the PUB action, the probability to not publish a content varies from a user to another according to his trust level.

The trust quality of a video varies according to the content it holds, which can vary from good to untrusted, or even unauthorized content. For this reason, the videos' trust should be divided into a set of levels according to their contents. Let K denote the number of possible trust levels. The quality of the video levels is in a decreasing order, such that the first level consists of videos with the highest trust value while videos in level $\mathcal{L}$ consists of videos with a lower trust value. As depicted in Fig. 3, the states space can be defined as follows:

$$\mathcal{S} = (\mathrm{UP}, \mathrm{SR}, \mathrm{HR}, \mathrm{PB}, \mathrm{NP}, \mathrm{AN}_1, \cdots \mathrm{AN}_K, \mathrm{SA}_1, \cdots \mathrm{SA}_K) \tag{8}$$

where,

- UP: a state that refers to the upload of a video from a user;
- $\mathrm{AN}_1, \cdots \mathrm{AN}_K$: a set of states where the analytical checking is performed. AN_i refers to the state where the analytical checking is performed for a video with a trust level i. The transition probability (TP) from state UP to state AN_i varies according to the trust level of the user owning the video in question;
- $\mathrm{SA}_1, \cdots \mathrm{SA}_K$: a set of states where the analytical checking step is skipped. SA_i refers to the state where the analytical checking is skipped for a video with a trust level i. Similar to the previous states, the transition probability from state UP to state SN_i varies according to the trust level of the user owning the video in question;
- SR: a state that refers to the soft review, whereby the uploaded video is reviewed and is more likely to be published. A video with a high trust value is more likely to be $i)$ directly published or $ii)$ go to the state SR, then from that state, it will be published;
- HR: this state refers to the hard review, whereby the uploaded video is reviewed and is more likely not to be published. A video with a low trust value is more likely $i)$ not to be published or $ii)$ go to the state HR, then from that state, it will not be published;
- PB: this state refers to publishing the video;
- NP: this state refers to not publishing the uploaded video.

For the sake of simplicity, to present the different transition probabilities, we assign an integer number for each state. As depicted in Fig. 3, while the state UP is numbered 0, the state NP is numbered 2K + 4. Each state $i \in \{\mathrm{AN}_1, \cdots \mathrm{AN}_K\}$ is numbered by i, while each state $i \in \{\mathrm{SA}_1, \cdots \mathrm{SA}_K\}$ is numbered by $K + i$. The states PB, SR and HR are numbered $2K + 1$, $2K + 2$, $2K + 3$, respectively.

In our DMDP, each state $s \in \mathcal{S}$ is mapped with a set of possible actions $\mathcal{A}_s \in \mathcal{A}$. We denote by $p(s'|s, a)$ the transition probability from a state s to a state s' when action $a \in \mathcal{A}_s$ is used. Moreover, each state $s \in \mathcal{S}$ has a specific reward $r(s)$ that can be defined according to: $i)$ the cost in curred in terms of analytical and reviewing processes; $ii)$ the impact on the trust level of that user and the network. Formally, DMDP is defined as follows:

$$(\mathcal{S}, \mathcal{A}, (\mathcal{A}_s, s \in \mathcal{S}), (p(s'|s, a), (s, s') \in \mathcal{S}^2), (r(s), s \in \mathcal{S}))$$

Let P denote the transition probabilities matrix between different states. P is mainly affected by the trust value of the network and the user who uploads the video. The transition probabilities from the state UP to the states AN_i and SA_i should be the same. This is due to the fact that either performing or skipping the analytical checking will not affect the quality of the uploaded video. Formally, $P(\mathrm{AN}_i/\mathrm{UP}, a) = P(\mathrm{SA}_i/\mathrm{UP}, a)$ for $a = \mathrm{UPL}$ and $\forall i \in \{1, \cdots K\}$. The higher the trust value of a user is, the higher the trustworthiness of his/her videos is. For this reason, in case of a user whose trust value is high, then $P(\mathrm{SA}_{i-1}/\mathrm{UP}, \mathrm{UPL}) \geq P(\mathrm{SA}_i/\mathrm{UP}, \mathrm{UPL})$ and $P(\mathrm{AN}_{i-1}/\mathrm{UP}, \mathrm{UPL}) \geq P(\mathrm{AN}_i/\mathrm{UP}, \mathrm{UPL})$ for $i \in \{2, \cdots K\}$. Inversely, in case of a user with low trust value, then $P(\mathrm{SA}_{i-1}/\mathrm{UP}, \mathrm{UPL}) \leq P(\mathrm{SA}_i/\mathrm{UP}, \mathrm{UPL})$ and $P(\mathrm{AN}_{i-1}/\mathrm{UP}, \mathrm{UPL}) \leq P(\mathrm{AN}_i/\mathrm{UP}, \mathrm{UPL})$ for $i \in \{2, \cdots K\}$.

The trust value of a user has also an impact on the transition probabilities between states $\mathrm{AN}_1, \cdots \mathrm{AN}_K, \mathrm{SA}_1, \cdots, \mathrm{SA}_K$ to the states $\{\mathrm{SR}, \mathrm{HR}, \mathrm{PB}, \mathrm{NP}\}$. The higher the trust value of a user is, the higher the transition probability to the states SR and PB becomes. Also, an increase in the trust value has a higher positive impact on the transition probabilities to state PB than the transition probabilities to state SR, such that the transition probability could equal one for the user with the highest trust value. Meanwhile, for a user with a low trust value, the states HR and NP should receive higher probabilities.

The transition probability to state NP is higher than the TP of state HR when the trust value is lower, such that the TP to state NP could equal one for the user with the lowest trust value. Moreover, the TP from states SR and HR to states PB and NP are also affected by the trust values of the user.

The higher the trust level of a user is, the higher the transition probabilities from states SR and HR to state PB become. This can be explained as follows, a user with a high trust level is more likely to upload authenticated videos, and it is more probable that the feed-backs from the reviewers will be positive.

Meanwhile, the lower the user's trust level is, the lower the transition probabilities from states SR and HR to state PB become.

The reward $r(s)$ at each state $s \in \mathcal{S}$ is affected by different parameters, such that the trust of the network, the users, as well as the cost, in terms of resources spent for analytical and reviewer checking at each state. When a user uploads a video, the reward of that state equals to a positive number λ. Formally, $r(UP) = \lambda$, such that $\lambda \geq 0$. Let μ denote the amount that the system can gain from uploading a video. Let δ denote the cost needed to perform the analytical checking of the video content. The value of μ could be fixed according to the efficiency of the underlying algorithm and the cost of the server used for performing the task.

For each state $s \in \{SA_1, SA_2, \cdots SA_{\mathcal{L}}\}$, where the analytical checking of the video content is skipped, the reward is $r(s) = \mu$.

Meanwhile, for each state $s \in \{AN_1, AN_2, \cdots AN_K\}$, where the analytical checking is performed, thus the reward is $r(s) = \mu - \delta$. For the states SR, HR, the reward is defined according to the fee that the system is willing to pay for each reviewer. Both states have the same reward, which is denoted by β. Meanwhile, the reward of the states PB and NP is defined according to the trust level of each user. The higher the trust level of a user is, the higher the reward of the state PB becomes. While the reward of the state NP inversely increases with the trust level of users, the trustier a user is, the smaller the reward value of the state NP becomes. Let θ denote the reward of the state PB, while ϑ denotes the reward of the state NP. Note that the reward of the state PB is significantly higher than the reward of the state $i \in \{SA_1, \cdots AS_K\}$. Moreover, the reward of a state $i \in \{SA_1, \cdots AS_K\}$ is higher than the reward of a state $j \in \{AN_1, \cdots AN_K\}$. Furthermore, the reward of any state $i \in \{AN_1, \cdots AN_k, SA_1, SA_k, SR, HR, PB\}$ is significantly higher than the reward of the state NP. Hereunder, we summarize the rewards of the different states:

$$r(s) = \begin{cases} \lambda & s = \mathrm{UP} \\ \mu - \delta & s \in \{\mathrm{AN}_1, \cdots \mathrm{AN}_K\} \\ \mu & s \in \{\mathrm{SA}_1 \cdots, \mathrm{SA}_K\} \\ \beta & s \in \{\mathrm{SR}, \mathrm{HR}\} \\ \theta & s = \mathrm{PB} \\ \vartheta & s = \mathrm{NP} \end{cases} \tag{9}$$

where $\delta \geq 0$ and $\theta >> \mu >> \vartheta$.

Let T denote the number of epochs that are executed when a video is uploaded. Let $\pi = \{\pi_1, \cdots \pi_T\}$ denotes the sequence of decisions taken at all the epochs. Given the initial state s = UP and a discount factory $\gamma \in]0, 1]$, the expected discount reward of the policy $\pi = \{\pi_1, \cdots, \pi_T\}$ is given as follows:

$$V_\gamma^\pi = \lim_{T \to \infty} E_\gamma^\pi \left\{ \sum_{t=1}^{T} \gamma^{t-1} r_t \right\} \tag{10}$$

where r_t denotes the reward received at the epoch t.

Let $V(s)$ denote the maximum discount total reward given the initial state s. In this case, $V(s) = \max_{\pi \in \Pi} V^\pi(s)$. From [37], the optimal equations are given by:

$$V(s) = \max_{\pi \in \Pi} \left\{ r(s) + \sum_{s' \in S} \gamma P(s'|s, a) v(s') \right\} \tag{11}$$

The solutions of the equations correspond to the maximum expected discount total reward $V(S)$ and the optimal policy $\pi^*(s)$. Formally, $\pi^*(s)$ is defined as follows:

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} \sum_{s' \in S} P(s'|s, a) V^*(s') \tag{12}$$

where $\pi^*(s)$ indicates the optimal decision that should be taken at each state. There are several algorithms that can be used for solving the optimization problem given by Equation (10). Value

TABLE I
VIDEOS DATA

Parameters	Video 1	Video 2	Video 3	Video 4	Video 5	Video 6	Video 7	Video 8	Video 9	Video 10
Duration (seconds)	60	80	90	85	100	93	83	75	120	110
Size (MB)	8,2	11,7	22	7,6	8,2	9,4	7,1 MB	11	15,9	8,7
Number of frames	1800	1920	2159	2548	2398	2789	2484	2249	2879	2638
Quality (pixel)	720	720	1080	360	360	480	360	1080	720	360
Analyze time	244	242	330	279	278	330	280	378	371	297

iteration and policy iteration are two notable examples. Eqn (13) as shown at the bottom of this page.

V. Experimental and Simulation Evaluation

In this work, we have benchmarked the video analysis service and evaluated the performance of the decision algorithm. Our virtual environment is set up on a KVM hypervisor in a dual Intel E3-1231. In this setup, we used a virtual machine (VM) deployed on top of a bare-metal server that runs Ubuntu server 16.04 LTS as operation system. The VM uses 8 Vcores CPU and 32GB memory. The rest of this section is organized as follows. First, we describe the real testbed experiments for benchmarking the required time for analytically checking the content of videos with different sizes and durations. Second, we explain the benchmark results obtained from the testbed experiments.

A. Benchmarking the Video Analysis Service

The impact of the video analysis service is evaluated in terms of time needed to analyze an uploaded video. In this benchmarking, we used the VM described above. In order to get the time needed to analyze a video, we created a set of videos with different duration and qualities. These videos were sent to the video analyzer as input, which is running on a VM. The video analyzer goes through every frame composing the video, applies object recognition methods to catch all objects detected and collects all words recognized. The softwares used to analyze these videos are *Tensorflow* trained with our dataset and *OpenCv*. At the end of each video processing, we store the time needed to analyze the video, the quality, the number of frames, the duration, the size, the words, and the objects detected. Table I shows the details about some analyzed videos. The data collected on this experiment include the analyzing time, the objects detected and the words found. This experiment also helps us to estimate the efforts required for performing the analytical checking of each video.

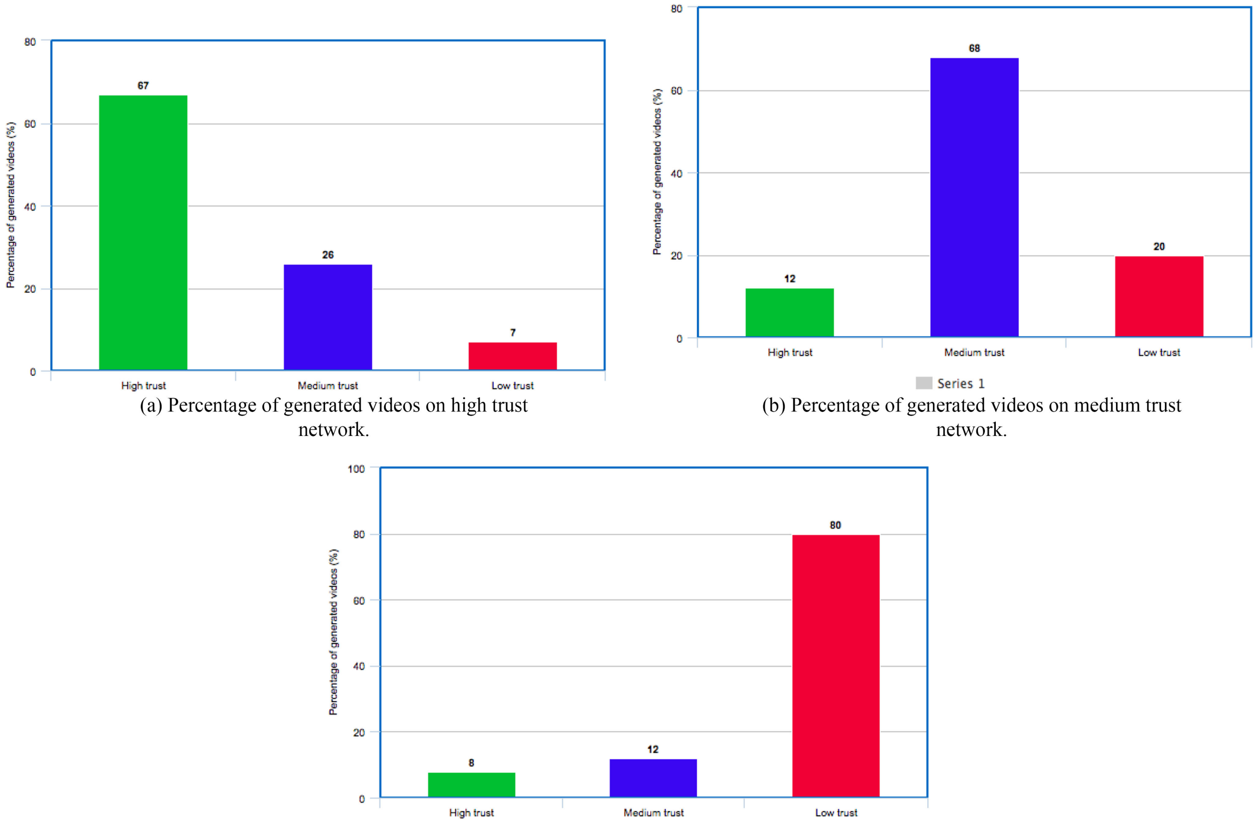
B. Performance of VUDP

The proposed decision algorithm at the VUDP model is evaluated in terms of the following metrics:

- The time required for analyzing the uploaded videos. This metric shows the overheads of the proposed solution in terms of processing time and resources consumption;
- The percentage of high trusted videos published in the network. This metric shows the positive impact of the proposed solution for publishing videos in the network, which will have a positive impact on the trust value of the network;
- The percentage of medium trusted videos published in the network. This metric also shows the positive impact of the proposed solution on the trust value of the network;
- The percentage of low trusted videos published in the network. This metric shows the false positive decisions taken by the proposed algorithm. An increase in the number of videos published with low trust will have a negative impact on the trustworthiness of the network.

We have evaluated the behavior of the proposed algorithm in three different scenarios as depicted in Fig. 4. The first scenario considers a lowly trusted network (LT), whereby most of the generated videos are with low trusted value. In this case, the trust value of the video is generated from the range [0%, 30%], such that 100% is the highest trust value that a video can receive. The second scenario considers medium trusted (MT) network, whereby the trust value of these videos is selected from the interval [30%, 60%]. Last but not least, the third scenario considers the network with high (HT) trust value. The trust values of these videos were selected from the interval [60%, 100%]. We conducted two sets of experiments: i) first, we fixed the number

$$Pr_{ij} = \begin{cases} p & \text{i = 0, j = 1} \mid \text{j = k + 1} \\ p + \frac{2(j-1)(1-kp)}{k(k-1)} & i = 0, j \in [2, k] \\ p + \frac{2(j-k-1)(1-kp)}{k(k-1)} & i = 0, j \in [k+2, 2k] \\ p \times \sigma & i \in [1, 2k], j = 2k+1 | j = 2k+3; \sigma \in]0, \frac{1}{p}] \\ 1 - p \times \sigma & i \in [1, 2k], j = 2k+2 | j = 2k+4 \\ \alpha & (i = 2k+2, j = 2k+1) | (i = 2k+3, j = 2k+4); \alpha > \frac{1}{2} \\ 1 - \alpha & (i = 2k+2, j = 2k+4) | (i = 2k+2, j = 2k+1) \\ 1 & i = 2k+1 | i = 2k+4, j = 0 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$



(a) Percentage of generated videos on high trust network.

(b) Percentage of generated videos on medium trust network.

(c) Percentage of generated videos on low trust network.

Fig. 4. Percentage of generated videos on each network type.

of videos to 15 per user while varying the number of users from 0 to 300; $ii)$ second, we fixed the number of users composing the network to 100 while varying the number of videos per user from 0 to 150.

To derive the VUDP policy, we used a python implementation of the policy iteration algorithm contained in the package *mdp-toolbox*. We also set the number of trust level *k* to 3, accordingly the number of states in DMDP is 11. The transition probability is generated by the equation 13, while *i* represents the starting state and *j* denotes the arrival state. The value of *p* is computed from the normal distribution probability representing the trust level of users.

The probability to transit from the state *UP* to $\{AN_1\}$ or $\{SA_4\}$ is *p*. We can easily deduct that the probability to transit to the states $\{AN_2\}$ and $\{AN_3\}$ decreases or increases according to the level of trust of the given user. In the same way, the probability to transit to states $\{SA_5\}$ or $\{SA_6\}$ decreases or increases. For the sake of simplicity, to compute those transition probabilities, we defined a numerical sequence for the actions *ANC* and *SANC*, with the first element of that sequence equals to *p*. We know that the sum of transition probabilities for a given action equals to 1; then the other transition probabilities are computed by using the numerical sequence proprieties defined as follows:

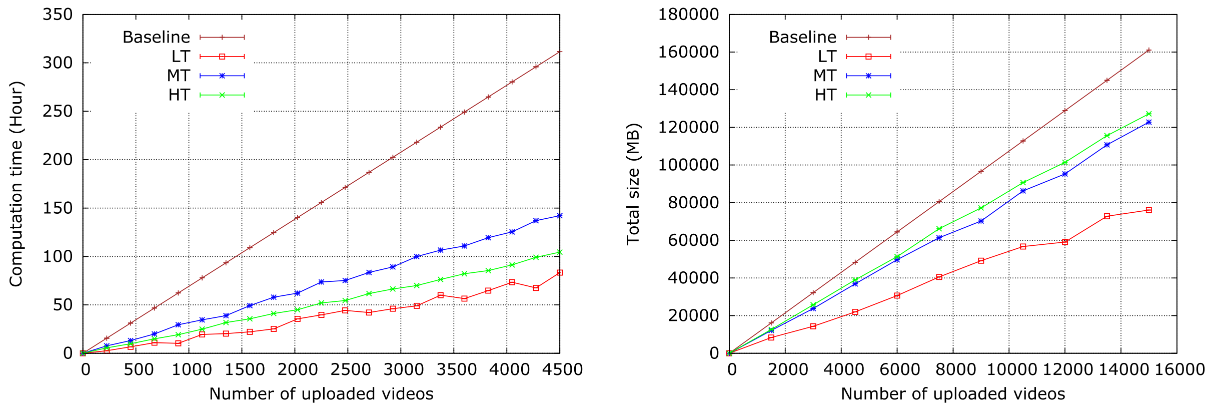
$$S_n = \begin{cases} p & \text{n = 1} \\ p + (n-1)R & \text{otherwise} \end{cases} \quad (14)$$

with:

$$\sum_{i}^{3} S_i = 1 \quad (15)$$

In Fig. 5, we compared the proposed algorithm to a baseline solution, whereby all uploaded videos are analytically checked. Fig. 5(a) shows the performance of our decision algorithm in terms of computational time. Meanwhile, Fig. 5(b) shows the performance of the decision algorithm in terms of the size of videos successfully uploaded. In this figure, the increase in the number of uploaded videos has a positive impact on system utilization and a negative impact on the cost. Formally, an increase in the size of videos increases the number of resources needed to process the uploaded videos, and hence the cost will be negatively affected. In contrast to the decision algorithm, the baseline solution, as shown in Fig. 5(b), uploads all videos to the system.

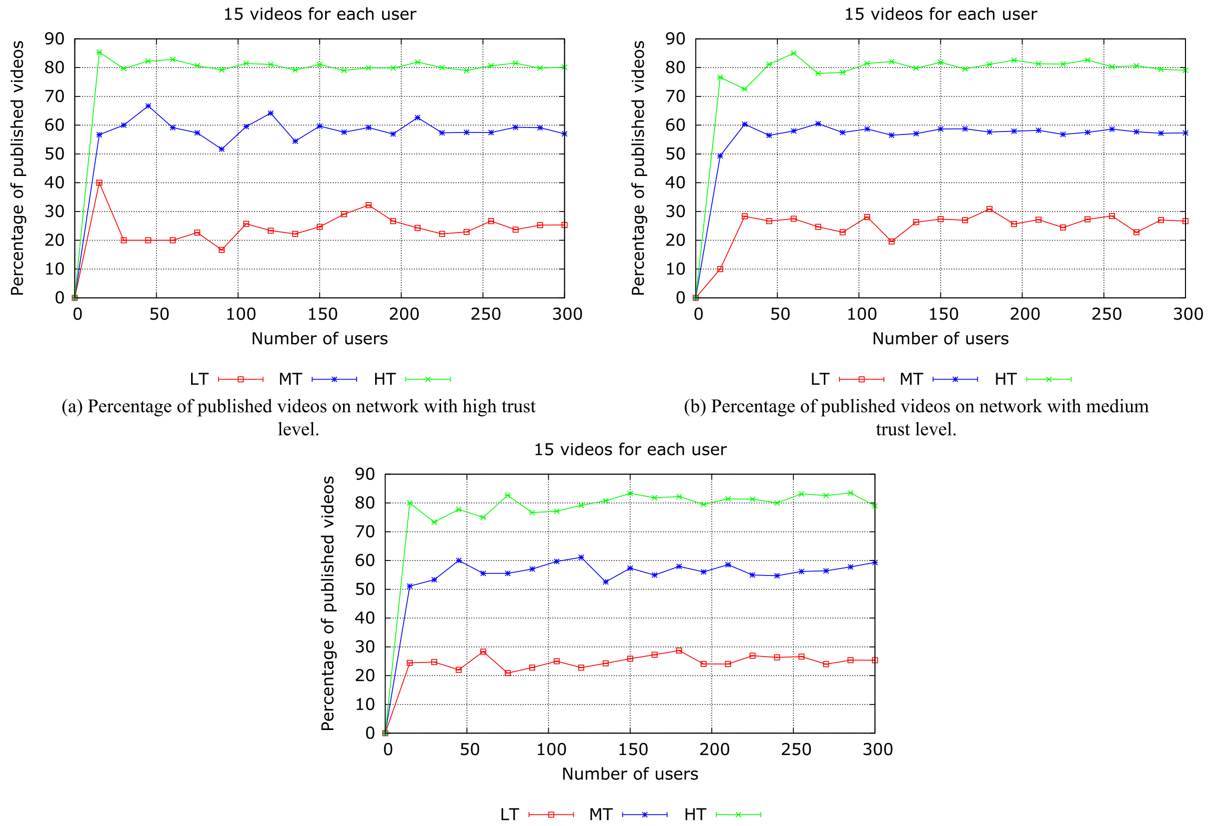
From Fig. 5(a), the first observation that we can draw is that regardless the scenario, the proposed algorithm outperforms the baseline solution. While the baseline solution performs the analytical checking of all videos, the decision algorithm performs analytical checking only for a few numbers of videos. From this figure, we also observe that the computation time of the baseline solution is largely greater than the one of the decision algorithm and that is for any trust level. For instance, regardless the trust value of the uploaded videos, our decision algorithm does not



(a) Performance of the proposed solution in terms of analytical checking overhead.

(b) Performance of the proposed solution in terms of storage.

Fig. 5. Performance of the proposed solution as a function of the analytical checking overhead and storage.



(a) Percentage of published videos on network with high trust level.

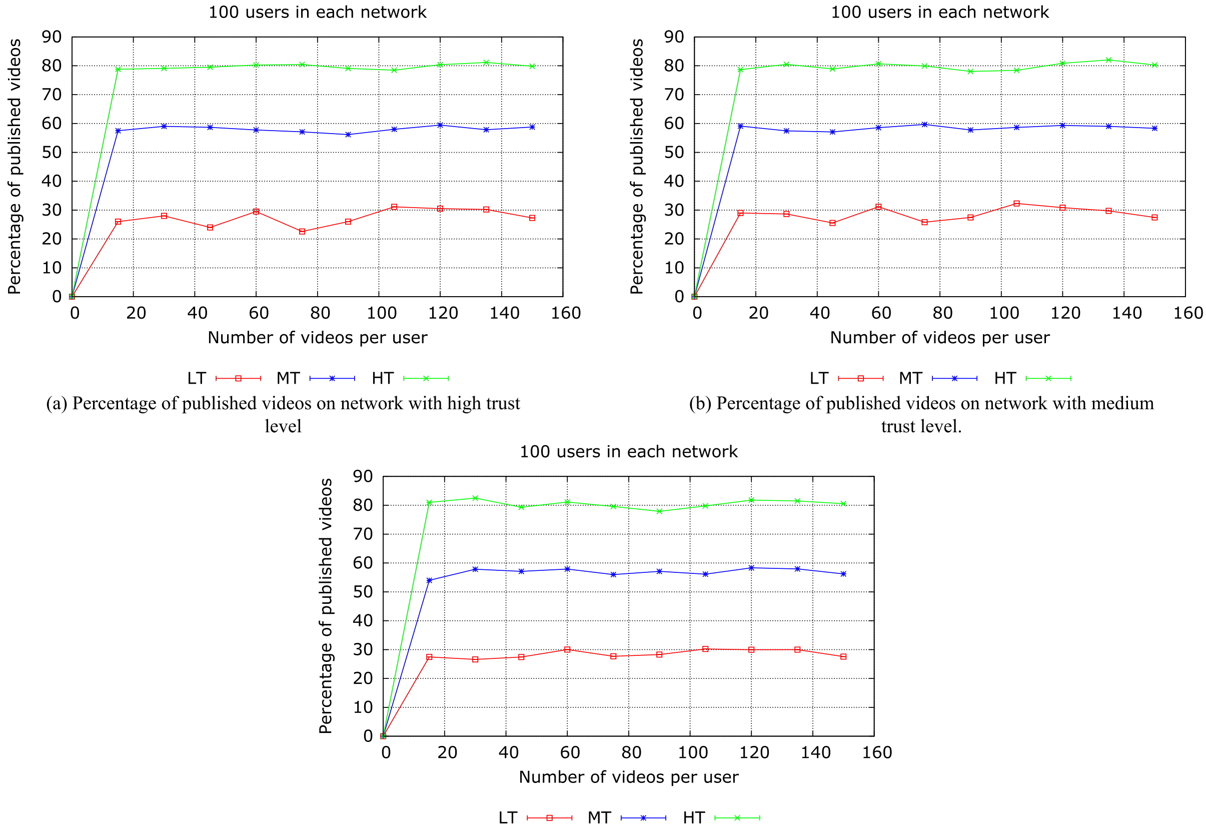
(b) Percentage of published videos on network with medium trust level.

(c) Percentage of published videos on network with low trust level.

Fig. 6. Performance of the proposed solution as a function of the number of users.

take more than 150 hours to analyze 4500 videos, whereas the baseline solution spends more than 300 hours for analyzing those videos.

From Fig. 5(b), we observe that the baseline solution stores more videos compared to the envisioned decision algorithm. While the baseline solution stores any uploaded videos, the decision algorithm accepts the upload of the videos that have only a high trust value. This leads to reducing the cost and preventing users from uploading insecure, untrusted and unauthorized contents. From this figure, we observe that the baseline solution stores more videos than the decision algorithm regardless the underlying scenario. Also, we observe that the envisioned decision algorithm stores more videos in the high trust level scenario than the medium and low trust level scenarios. Moreover, the envisioned decision algorithm stores more videos in the medium level scenario than the low-level scenario. This can be explained by the fact that, in the high-level scenario, we have more videos and users with high trust values than in the other scenarios, thus the decision algorithm stores more videos. The same thing is observed in case of the medium trust level scenario whose videos



(a) Percentage of published videos on network with high trust level

(b) Percentage of published videos on network with medium trust level.

(c) Percentage of published videos on network with low trust level.

Fig. 7. Performance of the proposed solution as a function of the number of videos.

have higher trust values than the ones of the low-level scenario. This figure demonstrates the efficiency of the proposed solution in saving the cost and preventing the manipulation of insecure, untrusted and unauthorized contents.

Fig. 6 and Fig. 7 show the impact of the number of users and videos on the percentage of published videos with high, medium and low trust levels, respectively. Both figures show similar performances in terms of trust levels. From Fig. 6 and Fig. 7, we observe that the trustworthiness level of the network has a positive impact on publishing the videos with high trusted values. In all the scenarios, we can notice that almost 85% of the videos with high trust values and about 60% of the videos with medium trust were published, while only 30% of the videos with low trust level have been published. In all scenarios, whether it is high, medium or low trust networks, the DMDP model is optimized in a way to maximize the publication of the videos with high and medium trust levels, while minimizing the publication of the videos with low trust level.

VI. Conclusion

Social multimedia networks are gaining a lot of momentum and their services are becoming the most popular ones among the community of Internet users. The data generated and exchanged by users of these networks become diverse. They include videos, documents, text, and pictures. Unfortunately, there are users that can insert insecure, untrusted and unauthorized contents. Thus, there is need for an effective way to control and verify the exchanged content. In this work, we focused on how to ensure that the users upload only secured, trusted and authorized videos to the social multimedia networks. We therefore proposed a complete framework that takes into account different aspects to attribute trust values to both users and content and to accordingly secure video streaming. The proposed framework has been designed in a way to reduce the resources utilization in terms of CPU, RAM, and storage. Moreover, we proposed a video uploading decision process module that leverages the historical behaviors of users for making the right decisions on either allowing or denying the upload of videos. This module uses an infinite discrete Markov decision process (DMDP) for taking those decisions. Also, this module can decide for either to analytically check the contents or send them to external reviewers before publishing them or forbidding their publication. The simulation results demonstrate the efficiency of the proposed algorithm in terms of publishing the good contents and forbidding the bad ones. Also, the simulation results demonstrate the efficiency of proposed algorithms in terms of minimizing the incurred computational cost.

References

[1] L. Gao *et al.*, "A popularity-driven video discovery scheme for the centralized P2P-VoD system," in *Proc. 8th Int. Conf. Wireless Commun. Signal Process.*, Oct. 2016, pp. 1–4.

[2] W. Chang and J. Wu, "Social VoD: A social feature-based P2P system," in *Proc. 44th Int. Conf. Parallel Process.*, Sep. 2015, pp. 570–579.

[3] T. Taleb, N. Kato, and Y. Nemoto, “Neighbors-buffering-based video-on-demand architecture,” *Signal Process., Image Commun.*, vol. 18, no. 7, pp. 515–526, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596503000390

[4] T. Taleb and N. Taleb, “System and method for creating multimedia content channel customized for social network,” Patent PCT/US2011/049 159, Nov. 2011.

[5] Statista. Social media usage worldwide. [Online]. Available: https://www.statista.com/study/12393/social-networks-statista-dossier/. Accessed on: Jul. 20, 2018.

[6] G. Noh, H. Oh, K. H. Lee, and C. K. Kim, “Toward trustworthy social network services: A robust design of recommender systems,” *J. Commun. Netw.*, vol. 17, no. 2, pp. 145–156, Apr. 2015.

[7] T. Taleb, A. Ksentini, M. Chen, and R. Jantti, “Coping with emerging mobile social media applications through dynamic service function chaining,” *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2859–2871, Apr. 2016.

[8] T. Taleb and A. Ksentini, “Impact of emerging social media applications on mobile networks,” in *Proc. IEEE Int. Conf. Commun.*, Jun. 2013, pp. 5934–5938.

[9] L. Yang, Z. Zhang, W. Tian, and Q. Chen, “Advance on trust model and evaluation method in social networks,” in *Proc. 6th Int. Conf. Genetic Evol. Comput.*, Aug. 2012, pp. 9–14.

[10] X. Liang, X. Lin, and X. S. Shen, “Enabling trustworthy service evaluation in service-oriented mobile social networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 310–320, Feb. 2014.

[11] M. K. Rahman and M. A. Adnan, “Dynamic weight on static trust for trustworthy social media networks,” in *Proc. 14th Annu. Conf. Privacy, Secur. Trust*, Dec. 2016, pp. 62–69.

[12] S. Hussain, N. Honeth, R. Gustavsson, C. Sandels, and A. Saleem, “Trustworthy injection/curtailment of DER in distribution network maintaining quality of service,” in *Proc. 16th Int. Conf. Intell. Syst. Appl. Power Syst.*, Sep. 2011, pp. 1–6.

[13] A. Ganz and A. Kumar, “A systems approach to teaching trustworthy computing,” in *Proc. 37th Annu. Frontiers Educ. Conf.*, Oct. 2007, pp. S1C–15–S1C–18.

[14] S. Hall, W. McQuay, and K. Littlejohn, “A trustworthiness evaluation framework for distributed networks,” in *Proc. IEEE Nat. Aerosp. Electron. Conf.*, Jul. 2012, pp. 51–56.

[15] S. Hall and W. McQuay, “Fundamental features of a unified trust model for distributed systems,” in *Proc. IEEE Nat. Aerosp. Electron. Conf.*, Jul. 2011, pp. 139–145.

[16] C. Jia, L. Xie, X. Gan, W. Liu, and Z. Han, “A trust and reputation model considering overall peer consulting distribution,” *IEEE Trans. Syst., Man, Cybern., Part A, Syst. Humans*, vol. 42, no. 1, pp. 164–177, Jan. 2012.

[17] K. Das and S. K. Sinha, “Essential pre-processing tasks involved in data preparation for social network user behaviour analysis,” in *Proc. Int. Conf. Intell. Sustain. Syst.*, Dec. 2017, pp. 28–32.

[18] R. Wang and G. Chen, “Mining negative links between data clusters,” in *Proc. IEEE Int. Conf. Commun. Problem-Solving*, Oct. 2015, pp. 520–523.

[19] W. Yuji, “The trust value calculating for social network based on machine learning,” in *Proc. 9th Int. Conf. Intell. Human-Mach. Syst. Cybern.*, Aug. 2017, vol. 2, pp. 133–136.

[20] G. Zhao, X. Qian, and X. Xie, “User-service rating prediction by exploring social users’ rating behaviors,” *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 496–506, Mar. 2016.

[21] Y. Tian, J. Srivastava, T. Huang, and N. Contractor, “Social multimedia computing,” *Computer*, vol. 43, no. 8, pp. 27–36, Aug. 2010.

[22] J. Sang and C. Xu, “On analyzing the ‘variety’ of big social multimedia,” in *Proc. IEEE Int. Conf. Multimedia Big Data*, Apr. 2015, pp. 5–8.

[23] F. Jabeen, Z. Hamid, A. Akhunzada, W. Abdul, and S. Ghouzali, “Trust and reputation management in healthcare systems: Taxonomy, requirements and open issues,” *IEEE Access*, vol. 6, pp. 17246–17263, 2018.

[24] T. Phukseng and S. Sodsee, “Calculating trust by considering user similarity and social trust for recommendation systems,” in *Proc. 12th Int. Conf. Intell. Syst. Knowl. Eng.*, Nov. 2017, pp. 1–6.

[25] E. K. Wang, Y. Li, Y. Ye, S. M. Yiu, and L. C. K. Hui, “A dynamic trust framework for opportunistic mobile social networks,” *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 1, pp. 319–329, Mar. 2018.

[26] T. DuBois, J. Golbeck, and A. Srinivasan, “Predicting trust and distrust in social networks,” in *Proc. IEEE 3rd Int. Conf. Privacy, Secur., Risk Trust, IEEE 3rd Int. Conf. Social Comput.*, Oct. 2011, pp. 418–424.

[27] K. Zolfaghar and A. Aghaie, “Mining trust and distrust relationships in social web applications,” in *Proc. IEEE 6th Int. Conf. Intell. Comput. Commun. Process.*, Aug. 2010, pp. 73–80.

[28] X. Wang, Z. Xu, X. Xia, and C. Mao, “Computing user similarity by combining simrank++ and cosine similarities to improve collaborative filtering,” in *Proc. 14th Web Inf. Syst. Appl. Conf.*, Nov. 2017, pp. 205–210.

[29] T. DuBois, J. Golbeck, and A. Srinivasan, “Rigorous probabilistic trust-inference with applications to clustering,” in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. Intell. Agent Technol.*, Sep. 2009, vol. 1, pp. 655–658.

[30] L. Cao and Y. Jiang, “An effective background reconstruction method for video objects detection,” in *Proc. 3rd Int. Conf. Netw. Distrib. Comput.*, Oct. 2012, pp. 161–165.

[31] B. Lu, S. Zhu, X. Ju, and L. Chen, “Adaptive codebook modeling based multiple objects detection,” in *Proc. Chinese Control Decis. Conf.*, Jun. 2018, pp. 2471–2475.

[32] F. Jabloncik, L. Hargas, D. Koniar, J. Volak, and Z. Loncova, “Dynamic objects detection of the respiratory epithelium based on image analysis,” in *Proc. ELEKTRO*, May 2018, pp. 1–5.

[33] I. Agriomallos, S. Doltsinis, I. Mitsioni, and Z. Doulgeri, “Slippage detection generalizing to grasping of unknown objects using machine learning with novel features,” *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 942–948, Apr. 2018.

[34] S. Oh, M. Kim, D. Kim, M. Jeong, and M. Lee, “Investigation on performance and energy efficiency of CNN-based object detection on embedded device,” in *Proc. 4th Int. Conf. Comput. Appl. Inf. Process. Technol.*, Aug. 2017, pp. 1–4.

[35] B. E. Mada, M. Bagaa, and T. Taleb, “Efficient transcoding and streaming mechanism in multiple cloud domains,” in *Proc. IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.

[36] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2014.

[37] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed., New York, NY, USA: Wiley, 1994.

Badr Eddine Mada received the bachelor’s degree in mathematical and computer science and the master’s degree in software engineering from University Mohammed V, Rabat, Morocco, in 2014 and 2017, respectively. He is currently working toward the Doctoral degree with Aalto University, Espoo, Finland. His research focuses on mobile edge computing and open source networking.

Miloud Bagaa received the bachelor’s, master’s, and Ph.D. degrees from the University of Science and Technology Houari Boumediene Algiers, Bab Ezzouar, Algeria, in 2005, 2008, and 2014, respectively. He is currently a Senior Researcher with the Communications and Networking Department, Aalto University, Espoo, Finland. His research interests include wireless sensor networks, the Internet of Things, 5G wireless communication, security, and networking modeling.

Tarik Taleb received the B.E. degree (with distinction) in information engineering in 2001, and the M.Sc. and Ph.D. degrees in information sciences from Tohoku University, Sendai, Japan, in 2003, and 2005, respectively. He is currently a Professor with the School of Electrical Engineering, Aalto University, Espoo, Finland. He is the Founder and the Director of the MOSA!C Lab. He is the Guest Editor-in-Chief for the IEEE Journal on Selected Areas in Communications series on network softwarization and enablers.