Universität
Zürich^UZH

Muriel Figueredo Franco
Bruno Rodrigues
Burkhard Stiller

On the Recommendation of Protection Services

August 2019

IfI TECHNICAL REPORT — No. 2019.06

University of Zürich UZH
Department of Informatics IfI
Binzmühlestrasse 14, CH—8050 Zürich, Switzerland

M. F. Franco, B. Rodrigues, B. Stiller:
IfI Technical Report No. 2019.06, August 2019
Communication Systems Group CSG
Department of Informatics IfI
University of Zürich UZH
Binzmühlestrasse 14, CH—8050 Zürich, Switzerland
URL: http://www.csg.uzh.ch/

On the Recommendation of Protection Services

Muriel Figueredo Franco, Bruno Rodrigues, Burkhard Stiller
Communication Systems Group CSG, Department of Informatics IfI, University of Zürich UZH
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland
E-mail: [franco, rodrigues, stiller]@ifi.uzh.ch

***Abstract*—Cyberattacks are the cause of several damages on governments and companies in the last years. Such damage includes not only leaks of sensitive information, but also economic loss due to downtime of services. The security market size worth billions of dollars, which represents investments to acquire protection services and training response teams to operate such services, determines a considerable part of the investment in technologies around the world. Although a vast number of protection services are available, it is neither trivial for network operators nor end-users to choose one of them in order to prevent or mitigate an imminent attack. As the next-generation cybersecurity solutions are on the horizon, systems that simplify their adoption are still required in support of security management tasks.**
Thus, this paper introduces *MENTOR*, a support tool for cybersecurity, focusing on the recommendation of protection services. *MENTOR* is able to *(a)* to deal with different demands from the user and *(b)* to recommend the adequate protection service in order to provide a proper level of cybersecurity in different scenarios. Four similarity measurements are implemented in order to prove the feasibility of the *MENTOR*'s engine. An evaluation determines the performance and accuracy of each measurement used during the recommendation process.

Keywords – Cybersecurity, Recommender System, Protection Services.

I. Introduction

Cyberattacks determine a rising threat for governments, companies, and end-users. Beyond compromising the security and privacy of individuals, malicious attackers can negatively impact the economy of businesses supported by digital systems. Distributed Denial-of-Service (DDoS) attacks remain one of the most dangerous threats to service providers around the world. DDoS attacks are responsible for most occurrences impacting [4] service downtime and performance degradation. The growing number of unsecured Internet-of-Things (IoT) devices, for example, ease the spreading of botnets being able to launch massive attacks on service providers [11]. Although enormous DDoS attacks are the major cause of concern, cyberattacks at the application layer are evolving (*e.g.*, code injections and social engineering) and are equally dangerous to the targeted system [2]. As a response, efforts increased to develop the next-generation cybersecurity solutions (*e.g.*, based on artificial intelligence [25] and blockchain technology [19]).

Currently, companies invest in protection services (*e.g.*, firewalls and anti-malware tools) and response teams to ensure availability and protect critical services and infrastructure. The cybersecurity market is worth billions of dollars [17] and investments are steadily rising. Thus, there are financial incentives for Protection Service Providers (PSP) to enter the market by offering protection services while end-users can reduce protection costs (*e.g.*, related to the deployment, configuration, and operation of services) by leveraging a competitive market for cybersecurity to meet their specific demands. These protections may include the acquisition of physical appliances, software licenses, virtual network functions, and cloud-based protection. Thus, although traditional models will still meet specific demands, a notable amount of next-generation protection services – as an instance of cybersecurity management – can adapt to flexible business models and provide a different level of protection on-demand.

Thus, there are a number of on-demand protection services and marketplaces available, which are not only offering protection services, but also offer alternatives regarding the deployment and management aspects of such services [6] [9]. However, it is not a trivial task for end-users to select one of them. Decision-making is even more critical when infrastructure is under attack and the decision to mitigate the attack should be provided on the basis of information about the infrastructure, such as its economic aspects, demands, and the characteristics of the attack. In this scenario, it is essential to observe not only how often attacks surpass the on-site infrastructure capacity, but also which off-site services can provide the necessary protection, considering their different service flavors, such as the amount of traffic supported, the capacity to address particularities of a determined attack, and price conditions. In this sense, recommender systems [23] provide a valuable security management tool to support decision during the detection and mitigation process.

Therefore, *MENTOR*, a protection service recommender system, is proposed as a support tool for cybersecurity management, being able to recommend services for the prevention and mitigation of cyberattacks. This work investigates similarity measure techniques to correlate information, such as budget constraints and the type of service required, from customers with different services available. Based on this, *MENTOR* is able to indicate an adequate service to protect infrastructures according to different demands, such as region, deployment time, and price conditions. Such services are based on state-of-the-art technologies, providing features to deliver, according to previous configurations, different levels of performance, security, and availability. In addition, an evaluation and discussion determine the performance and accuracy of each similarity measure technique implemented within *MENTOR*.

The remainder of this paper is organized as follows. Section II reviews the background and related work. Section III introduces *MENTOR* and provides details of the recommendation process. Section IV provides an evaluation regarding the effectiveness of the algorithms used in the recommendation process. Finally, Section V concludes the paper and recommends future work.

II. Background and Related Work

Recommender systems can be described as filters and techniques that provide suggestions to the user according to their needs [23]. Different approaches can be used to enhance the recommendation process, such as collaborative filtering using feedbacks from active users and content-based filtering based on a comparison between products and end-users' profile data. Such systems and approaches have been actively investigated by academia and industry for various purposes (*e.g.*, advertisement, entertainment, and shopping). [5] provides an overview of recommender systems and describes the advantages and limitations of different recommendation methods. Also, [3] survey on the current landscape of recommender systems and the techniques being used, thus highlighting the main challenges and opportunities for future directions.

Although recommender systems have been applied to different areas, such as advertising in vehicular networks [15] and location-based services recommendation [24], few works are investigating issues related to computer networks, such as cybersecurity issues and network economic perspectives. In the cloud computing area, [1] introduced the CSSR tool, which is a cloud service security recommender that identifies risks from different cloud computing models from the stakeholder's perspective. CSRR provides a comprehensive basis from which alternative security solutions are identified, based on specific stakeholder's needs. [18] provides a recommender system to predict cyberattacks by identifying attack paths and demonstrates how a recommendation method can be used to classify future cyber attacks. [14] introduced an interactive user interface for security analysts that recommends what data to protect, visualizes simulated protection impact, and helps build protection plans. However, these solutions do not directly address the recommendation of protection services.

In the context of mobile computing, [21] proposed DroidNet, a recommendation tool for permission control on Android devices. This tool provides a high confidence permission control recommendation based on feedback from end-users, who were using the same mobile applications. [28] proposed an approach for recommending apps by striking a balance between their popularities and the end-users' security concerns, thus, building a hash tree to efficiently recommended apps. Moreover, studies are focusing on privacy issues related to recommender systems in Internet-of-Things (IoT) scenarios [26] and on similarity measures to quantify proximity between the rank vectors in order to improve the accuracy of Wi-Fi indoor positioning [16].

However, it is important to note none of those solutions mentioned above directly tackle the recommendation of protection services (*e.g.*, cloud-based services [27] or Network Functions Virtualization (NFV) solutions [9]) to mitigate and protect against cyber attacks. Therefore, although past work investigated recommendation tools for the prediction of cyberattacks [10], there is a lack of solutions that establish an efficient path between victims and PSPs to deliver optimal solutions dealing with the rising number of cyberattacks efficiently. Thus, recommender systems, in such a context, can highly useful to reduce infrastructure damage, while reducing both Capital Expenditure (CAPEX) and Operational Expenditure (OPEX).

A. *Protection Services*

The market of protections services has grown together with the investments in cybersecurity. Nowadays, several PSPs are offering protections for different kind of attacks (*e.g.*, data leaks, DDoS and malwares) and demands. For example, [7] provides a repository that lists PSPs offering many protection services to address different cybersecurity threats, such as advanced threat protection, anti-virus, secure communications, and anti-phishing. The number of protections available is large and is growing in parallel with the investments in cybersecurity. Only on such a repository, one can obtain information and contract more than 80 protection services against DDoS attacks. Table I lists some of the DDoS attacks protection services available and its characteristics.

TABLE I: Example of DDoS Protection Service Providers [8]

Provider	**Service**	**Pricing**	**Deployment**
Cloudflare	Advanced DDoS Attack Protection	Free Trial	Cloud-based
Imperva	Incapsula	$5000 for setup $3500 per month	Cloud-based
Verisign	DDoS Protection Service	On-demand	Cloud-based
Arbor Networks	Arbor Cloud	On-demand	Cloud-based or On-site
Level 3 Communications	DDoS Mitigation	On-demand	Cloud-basedor On-site
Corero	SmartWall Threat Defense System	$1000 for setup $2500 per month	Cloud-based or On-site
Flowmon	DDoS Defender	Free Trial	Cloud based
Akamai	Kona Site Defender	$3000 for setup $6000 per month	Cloud-based
FENDE	Firewall and IDS	On-demand	NFV-based
SHIELD	IDS and Multi-layer Filters	On-demand	NFV-based

Besides the technical characteristics of the protection, different business models can be used as accounting and billing. Many PSPs are adopting the scheme of pay-as-you-go, which is preferred by customers with very specific demands, such as when an attack surpasses the infrastructure and a reaction is required fastly. In another direction, for continuous protection, there are business models based on a fixed price for the setup of the protection services and a fixed amount per month being paid to have the service running correctly. Others hybrid models are also available where customers can contract customized the service being contracted as well as the technical assistance.

Also, different technologies can be used to deliver the protection service for the costumers. One can have protection virtualized running directly on Cloud or as a Virtual Network Function (VNF) configured locally. Physical appliances and proprietary software's can also be acquired to be deployed directly on-site. The pros and cons of the different ways and technologies for deployment and cybersecurity management have been discussed exhaustively by academia and industry, which are, for example, taking into account: *i)* costs reduction for the deployment (both CAPEX and OPEX) and management of the services being contracted, *ii)* performance and availability concerning different scenarios (*e.g.*, capacity to support high-demands and resilience), and *iii)* simplicity and speed to acquire/configure services according to the customers demands. As this discussion are still open, researches that address the technical aspects regarding the enabler technology and its market are required in order to contribute to the development and broad adoption of the next-generation cybersecurity solutions.

III. *MENTOR* Solution

The *MENTOR* system assists network operators during the decision process on measures to protect critical infrastructure, thus performing an important security management task. For this, the recommender engine indicates protection services available from different PSPs to prevent and mitigate threats. *MENTOR* considers different properties from available protection services, the customer profile, and characteristics of the cyber attack to establish a fair recommender system, where one or more services from different PSPs (*e.g.*, both small companies and global players) can be proposed to neutralize a threat efficiently, while minimizing cost and reducing damage.

The process overview of *MENTOR* is depicted in Figure 1. One customer can describe his/her requirements (*e.g.*, budget, threat, and type of protection service) that can be used by the system to filter the content of available services from different PSPs in order to determine which one is most suitable to support all requirements and demands described. Different similarity algorithms are applied by the recommendation engine to determine the most appropriate service for the customer.

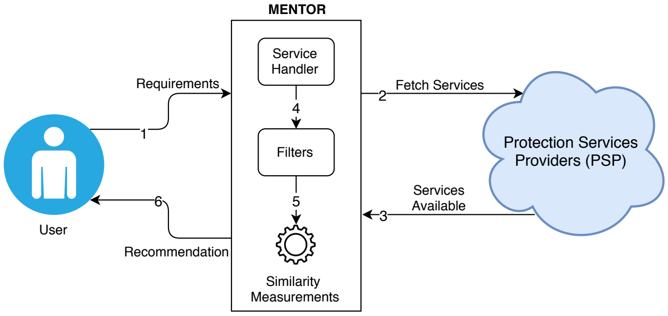


Fig. 1: Recommendation process overview

Figure 2 overviews the architecture of *MENTOR*. The recommendation flow is described as follows. First, in step 1, the *Service Requestor* receives information related to the infrastructure under attack and characteristics of the attack (*e.g.*, logs from monitors). Such information is transformed into an appropriate data structure and delivered to the *Extractor*, which initializes the recommendation process. Next, in the Extraction and Classification phases (steps 2 and 3), the information is analyzed and correlated with the type of attack in order to identify those requirements, which fend off the attack. In turn, a list of potential protection services is generated (step 4) and forwarded (step 5) to the recommendation engine. Finally, in step 6, the recommendation engine uses the customer profile input to define, which service from the list, is the optimal recommendation. Details about components that execute such actions in each step of the system are as follows.

In the first step, the *Service Requestor* receives data from monitors, stores relevant data in a database for future analysis, and, when a threat or imminent attack is identified, the component sends the significant information and meta-data to the *Extractor* component to start the recommendation process. Next, the *Extractor*, which is the first step of the recommendation process, is in charge of extracting relevant insights (*e.g.*, attackers, attack characteristics, and infrastructure impacts) from the data monitored. After the extraction, the information is forwarded to the data categorized into different kind of attacks.

During the next phase, the *Classifier* is responsible for classifying the extracted data according to the previously reported and identified attacks (*e.g.*, DDoS variations). To achieve this classification, techniques to identify attacks patterns and also a database providing attacks fingerprints [22] are applied. After the classification, the *Service Aggregator* communicates with different PSPs to obtain a list of available services available and relevant properties of each service (*e.g.*, price, type of service, and coverage area). Next, the database containing the services catalog is populated to supply customers. The list of PSPs can be modified according to customer preferences. Then, the *Retriever* is in charge of querying the *Service Aggregator*, who can fully or partially address the demands of the end-user. Such services selected and returned can yield different solutions targeting the same problem, but can vary in terms of performance, price, and the technology being used.

The final step of the recommendation process is composed by the *Recommendation Engine*, which supports different algorithms to select the optimal service, based on the list provided by the *Retriever*. Besides the input provided by the *Retriever*, a set of details is described by the customer to map the end-user profile and requirements. Therefore, to support such a decision, different aspects have to be considered, such as budget constraints, service coverage, and the capacity to address the particularities of an attack.

A. *Recommendation Engine*

The input data for the recommendation engine depicts a list of available protection services from PSPs. This list contains general information about the service (*e.g.*, price and type of service) as well as technical details regarding threats and attacks supported by each service. The data returned by each PSPs should optimally be provided through an interface (*e.g.*, RESTful API) to communicate with *MENTOR*'s *Service Aggregator* in order to be incorporated into the recommendation process. Thus, providing such an interface is in the interest of every PSP.

Table II presents those parameters that define the requirements of the end-user running the recommender system. These parameters are to be defined inside a profile and requirements descriptor (*e.g.*, a JSON file), containing useful information used during the filtering and recommendation steps conducted by the *Retriever* and the *Recommendation Engine*. One end-user, for instance, can use such descriptor to configure the recommender system to temporarily contract a reactive virtual protection service being remotely hosted in South America, with a deployment time of just a few seconds. The amount available to spend on such service will be defined as 500 US$. Also, if available, information about an imminent attack or threats possible to be exploited can be described. Thus, based on this information, protection services that do not support

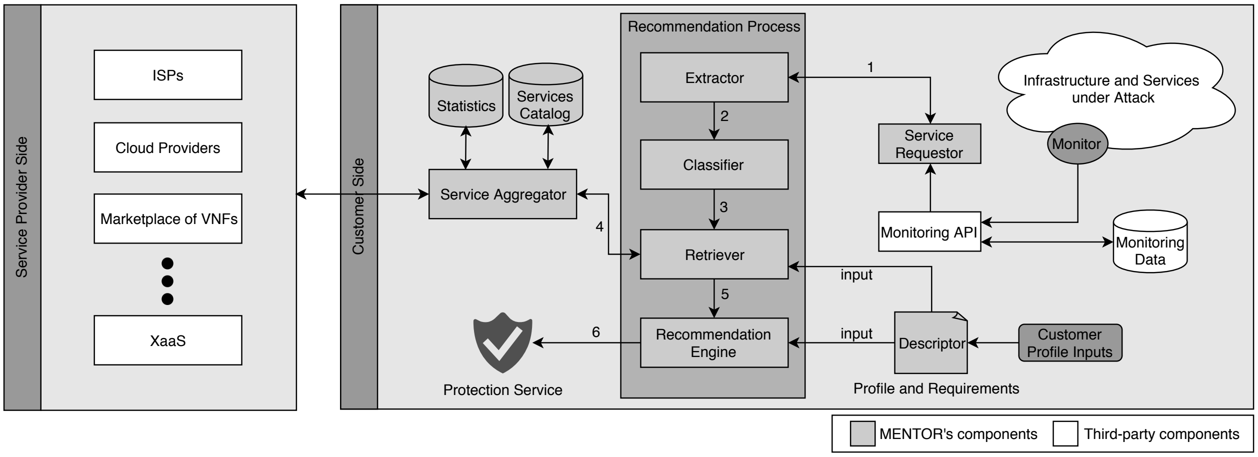


Fig. 2: The *MENTOR* architecture

all requirements will not be considered as a viable option. As the recommender system is able to adapt to different input scenarios, the descriptor can also be extended to support new parameters and relevant information provided by the protection services available, such as attack's behaviors or vulnerable applications.

TABLE II: Customer profile and requirements

Parameter	Description	Value
Type of Service	Describes if there is a demand to protect the network from further threats (*i.e.*, proactive) or react in order to mitigate imminent attacks (*i.e.*, reactive)	reactive or proactive
Type of Attack	Provides details of the attack which a protection is being required	*e.g.*, SYN Flood or a specific malware
Attack Details	Uploads log files or details about the attack	*e.g.*, DDoS fingerprints or behavior data of any attack
Region	Defines specific geolocalization that one protection service should be deployed or able to act	continent, country, city, or any
Deployment Time	Describes the maximum time between the service being contracted until it be able to protect the customer	seconds, minutes, hours, days, or any
Leasing Period	Defines the period for which the customer want to contract a protection service	minutes, hours, days, weeks, months, or any
Budget	The amount (*e.g.*, in Euro or USD) available to spend with protection	any

In order to evaluate the feasibility of the recommendation process, the *MENTOR* was assessed using four widely used similarity measures: *(i)* Euclidean distance, *(ii)* Manhattan distance, *(iii)* Cosine similarity, and *(iv)* Pearson correlation. These measures were selected because of their potential to quantify the similarity of two objects [23]. Thus, end-users demand can be compared with protections available in order to decide which fits better for each specific case. *MENTOR* was designed to be generic and extensible to support further algorithms to recommend protection services. In this regard, service requirements from customers and offered protection services are mapped as vectors in space as depicted in Figure 3, *i.e.*, their set of attributes as well as magnitudes represents a direction in space, allowing a geometric evaluation of similarity.

Equation 1 presents the Euclidean distance. The Euclidean distance is calculated by taking the square root of the sum of the squared pair-wise distances of every dimension. In terms of the recommendation process, a vector containing the parameters defined by the end-user (*cf.* Table II) are described as a vector x_i and each service available is transformed to a vector y_i in the same way. Then, the sum of differences of all individual squared pair-wise distances is square rooted. Thus, the Euclidean distance determines, if a service is adequate for the request: *i.e.*, the optimal recommendation is the service with the lowest possible Euclidean distance.

$$euclidean(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (1)$$

In a similar approach, the Manhattan distance, introduced in Equation 2, calculates the distance (β) between two vectors by considering the difference of the absolute values of each vector. The vector x represents the protection service and y the end-user profile. The best service is the one with the shortest diagonal path between the two vectors. Similar to the Euclidean distance, the protection service with the lowest possible value is optimal.

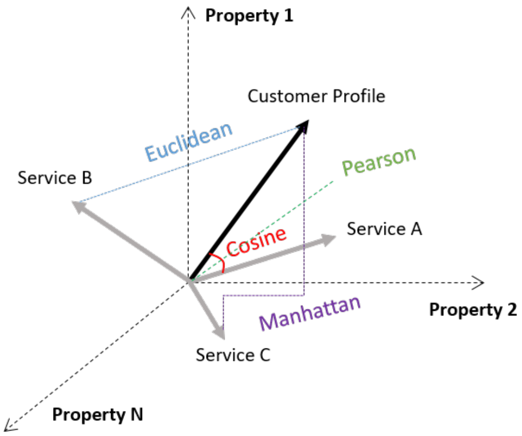


Fig. 3: Protection services mapped into vectors and compared to customer profiles using different similarity measures

$$manhattan(x, y) = \sum_{i=1}^{n} |x_i - y_i| \tag{2}$$

Equation 3 shows the Cosine similarity calculation, which finds the normalized dot product of two attributes x and y. $\cos(x, y)$, where x is any dimension of the end-user request and y is a dimension of a protection service), is calculated between the two vectors to decide, if one service fits the end-user request. If the angle is equal to 0°, the value for the cosine will be 1 (best recommendation) and it is less than 1 (*i.e.*, it ranges from 0.99 to -1) for any other angle.

$$\cos(x, y) = \frac{\sum_{i=1}^{n}(X_i \cdot Y_i)}{\sqrt{\sum_{i=1}^{n} X_i} \cdot \sqrt{\sum_{i=1}^{n} Y_i}} \tag{3}$$

The fourth measure under investigation is the Pearson correlation (*cf.* Equation 4). The Pearson correlation determines linear relationships between two normalized distributed variables. This correlation provides a value ranging from -1 to 1, representing the correlation between two vectors. Thus, the lower the value, the worse is a protection service x recommended for a demand y.

$$pearson(x, y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 (y_i - \overline{y})^2}} \tag{4}$$

The recommendation process works as depicted in Algorithm 1. Initial preparation steps involve *(a)* receiving/preparing input data and *(b)* filtering unrelated services. The input *(a)* requires receiving protection services from the *Service Aggregator*, which for the purpose of evaluation were created randomly. Next, the customer profile is determined (*i.e.*, received from the front-end or to establish the basis of comparison those services offered for protection. Before calculating similarities, unrelated services are discarded in the filtering process. This involves eliminating services, whose binary properties do not match the ones required by the customer, *e.g.*, type of service (reactive or proactive) or service region (Europe).

Step 1 involves the indexing of *(a)* service parameters required by the customer and *(b)* each service in order to build an integer array representing the service. These steps are necessary to map services and enable the application of similarity measures geometrically. Similarly, Step 2 is applied to each service to index its properties. Steps 3 and 4 involve the actual recommendation of services and storing of the rating. In Step 3, the customer profile is mapped as a vector Y and each protection services as a vector X, which are provided as input to similarity algorithms. In Step 4, ratings are stored as a similarity dictionary with the service ID as a key, especially to enable the export or plot similarity data later.

B. Prototype and Implementation

A prototype of *MENTOR* was implemented in order to evaluate the feasibility of such a solution practically. The web-based user interface was developed using ReactJS 16.8. The *Recommendation Engine* was implemented using Python 3.7.3. Flask 1.0.2 was used to implement REST APIs allowing the communication between components. The recommendation engine's code is available online [20].

Algorithm 1: *MENTOR's* recommendation algorithm

```
begin
    /* Receive services from Aggregator          */
    $services \leftarrow get\_services()$
    /* Receive customer requirements             */
    $profile \leftarrow get\_customerProfile()$
    /* Get customer index                        */
    $profile.index \leftarrow index(profile)$
    /* Step 1 - Filter Unrelated Services        */
    $services \leftarrow filter(services)$
    /* Step 2 - Recommend Remaining Services     */
    for $each\ s \in services$:
        /* Get index of each service             */
        $x \leftarrow index(services)$
        $y \leftarrow profile.index$
        /* Step 3,4 - Calculate and store
           similarities                          */
        $cosine[] \leftarrow s.cos \leftarrow cosine(x, y)$
        $euclidean[] \leftarrow s.euc \leftarrow euclidean(x, y)$
        $manhattan[] \leftarrow s.man \leftarrow manhattan(x, y)$
        $pearson[] \leftarrow s.pea \leftarrow pearson(x, y)$
end
```

End-users can access a dashboard provided to configure their requirements (*i.e.*, customer profile) and prioritizing each demand from *High* to *Low*. Defining priorities during the recommendation process, such as *High* priority for price, will impact the recommendation and, thus, returns the protection service with a lower price, while neglecting others, less prioritized criteria. After that, a list of the most recommended protection services available is returned. Even though a dashboard was implemented, the recommendation engine is loosely coupled to the dashboard and can be executed autonomously, without any interaction, only providing the adequate inputs (*e.g.*, attack's characteristics or specific demands) via *MENTOR*'s API in order to automate the process. The possible automation favors further steps towards a real-time recommendation of protection services.

As the *MENTOR* offers support for different algorithms, the recommendation algorithm can be selected by the end-user according to preference. In order to help in the decision process, different information (*e.g.*, graphs plots) are provided, representing how the algorithm classified each protection service. Thus, the end-user can visually process and understand the accuracy of a recommendation by comparing the vector describing the customer profile and the vector of each protection service.

To evaluate the dashboard's feasibility, the database was populated based on real-world protection services against DDoS attacks. Prices were generated randomly because most of these services do not publicly disclose prices. The *MENTOR* not only optimizes the service selection for end-users, but also encourages PSPs to actively publish their prices, which in turn increases price competition and usually results in a decreased price for the end-user. New services can be automatically added by using descriptors provided through the RESTful

API running on the PSPs side. For this, each PSP that wants to announce its service, needs to describe its services as a JSON file containing relevant information about the service and adhering to the model provided in Table II. After that, *MENTOR*'s components receives such descriptors and extract information to populate the database.

The prototype implemented also allows for an upload of log files to provide feedback on protection services. The end-users' feedback can be used to feed a reputation system for PSPs and customers. Thus, a reputation system can provide more accurate recommendations, decreasing the necessary trust placed in information advertised by the PSP. Reputation mechanisms are under development and there are still open challenges [13], such as how to verify and rely on the feedback data provided. The usage of a blockchain-based system can be further investigated.

Listing 1: Example of JSON file describing a customer profile

```
{
    budget: "200 USD",
    requirements:{
        protection_type: "Reactive"
        region: "Europe",
        deploymentTime:[
            "minutes"
        ],
        leasingPeriod:[
            "days"’
        ],
    }
    infrastructure:{
        technology: "Openstack"
        services_running:[
            "Apache Web Server",
            "MySQL Database"
        ],
        protection_running:[
            "IPtables"
        ]
        priority: "high"
    },
    attack:{
      type: "SYN Flood",
      log_file: "attack.pcap",
      fingerprint: "attack.json"
    }
}
```

By using the input of the end-user, a JSON file is automatically created via the dashboard, thus, describing requirements and attack characteristics. Also, information regarding the end-user's infrastructure can be described in order to refine *MENTOR*'s filters. For example, some protection services can be highly recommended for specific technologies (*e.g.*, Openstack-based infrastructure) , while other on-site protections (*e.g.*, IPtables-based Firewalls) are already running. This file can be created manually by any PSP, for example, following the standard defined by *MENTOR*. Listing 1 presents an example of JSON file describing a customer profile. This file is used as one input for the recommendation engine (*cf.* Figure 2) and, based on the requirements described on such a file, *MENTOR* can apply filters and determines the similarity between each service and the customer profile. The Algorithm 1 summarizes all steps of the recommendation process.

IV. Evaluations

The dataset generated for the evaluation contains 10,000 randomly generated protection services, such as each service was described based on parameters available for the customer profile (*cf.* Table II) and with a price range between 100 US$ and 1,000 US$. Thus, by using such data as an input to the *MENTOR*, the performance and accuracy of the measurement algorithms to recommend protection services were analyzed.

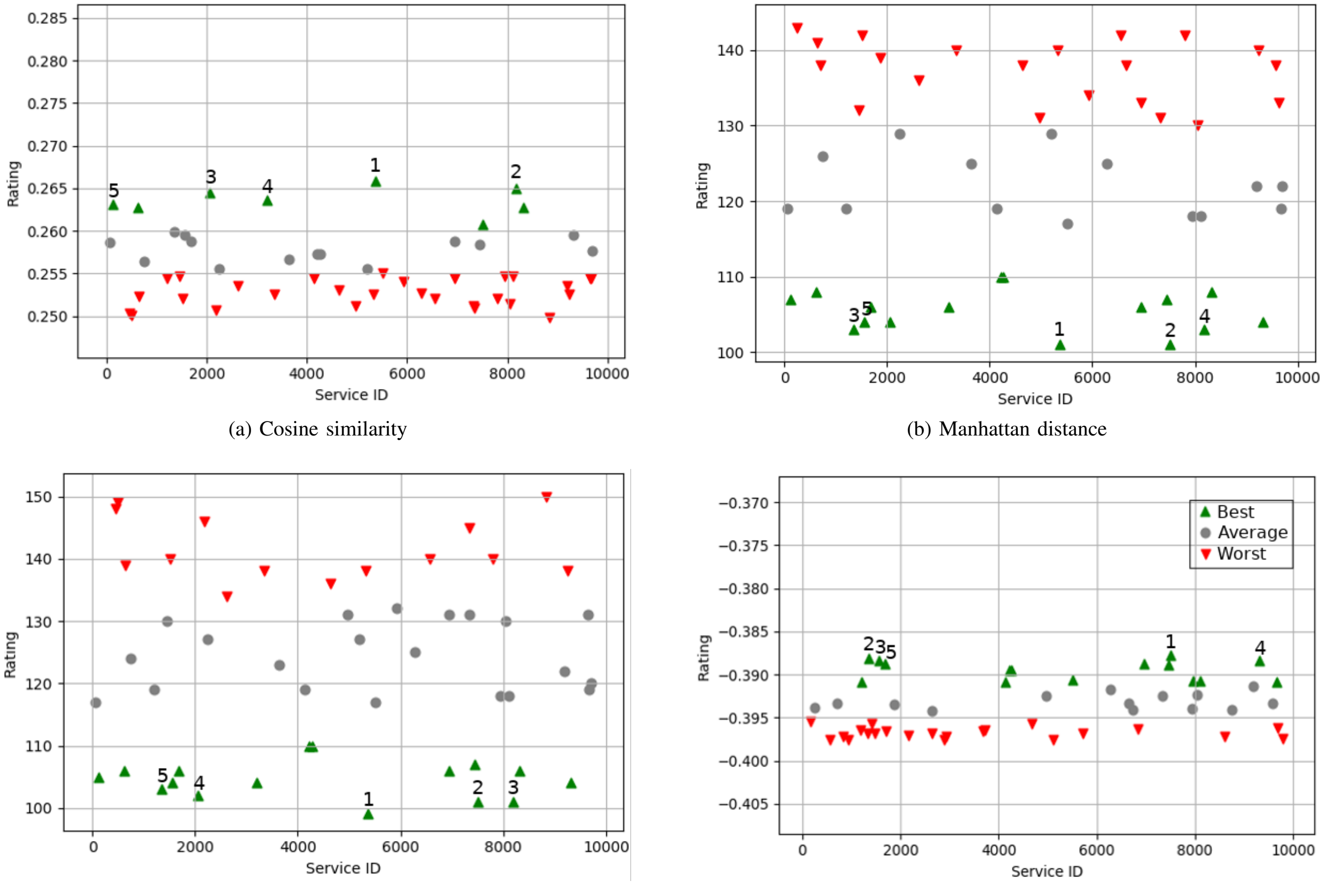
The four similarity measurements described beforehand were used to conduct this experiment. These requirements are indexed and translated into the vector composed by region, service type, deployment time, leasing period, and price, which is given as input to the recommendation engine. The customer profile (*i.e.*, input) was defined to represent a request for a reactive service against a DDoS attack, running in Europe with a deployment time in minutes, a leasing period in days, and the maximum budget to be up at 200 US$

After the dataset's creation and the customer profile input, the recommendation engine applies a filter to discard unrelated services (*e.g.*, outside the price range, region, or deployment time). The similarity is calculated based on the given vector (*i.e.*, customer profile) by using each algorithm available on the current version of the *MENTOR*.

Figure 4 depicts the top fifty ranked services for each similarity algorithm, in which the best five are highlighted in Table III. Although these recommended services were similar concerning the properties being compared, there are major differences in how these algorithms work depending on how the input vector is mapped. For example, all features of a protection service are described as a vector in space, in which certain properties can significantly change their direction, and consequently their rating. Therefore, high-magnitude variables (*e.g.*, price, deployment time, and leasing period) cause a major influence in the vector's direction in space, and thus, change the rating of its recommendation. For instance, a ”worse” rating can be given to services that, in practice, may be better than those specified in the customer profile. That is, a service with a slightly higher price and a significantly lower deployment period may have a worse ranking due to the disparity, in absolute terms, between the properties of the protection service.

This is observed in the distance-based algorithms (*e.g.*, Cosine, Euclidean, and Manhattan in Table III), in which the price was the most significant factor for the ranking of a service. For example, as observed in Figure 5, the service with ID 5362 was the service most similar to the vector specified by the customer profile (according to the distance-based algorithms), but it was not necessarily the best service. In this sense, services with a shorter deployment time (in the order of seconds) and without a significant price difference obtained a worse ranking due to the price difference. This happened for services ID 8182 and 7512 in the Tables of the Cosine, Manhattan, and Euclidean algorithms.

However, the major difference between the Pearson correlation and the distance-based algorithms is that it is invariant to the magnitude of elements. Hence, differences in service prices do not cause a major impact on their ratings because it mainly observes, whether properties of protection services and the customer profile vary in a similar way. Thus, the service



(a) Cosine similarity

(b) Manhattan distance

(c) Euclidean distance

(d) Pearson correlation

Fig. 4: Ratings of the fifty best-ranked protection services according to each algorithm.

TABLE III: Summary of the five best-ranked protection services according to ratings calculated as of Fig. 4

(a) Cosine similarity

Rank	ID	Rating	Price	Deployment	Leasing
1	5362	0.26585	100	Hours	Days
2	8182	0.26493	102	Seconds	Days
3	2062	0.26448	103	Seconds	Days
4	3202	0.26361	105	Hours	Days
5	122	0.26318	106	Seconds	Days

(b) Manhattan distance

Rank	ID	Rating	Price	Deployment	Leasing
1	5362	101	100	Hours	Days
2	7512	101	102	Seconds	Days
3	1352	103	104	Seconds	Days
4	8182	103	102	Hours	Days
5	1552	104	105	Seconds	Days

(c) Euclidean distance

Rank	ID	Rating	Price	Deployment	Leasing
1	5362	99.0202	100	Hours	Days
2	7512	101	102	Seconds	Days
3	8182	101.02	102	Hours	Days
4	2062	102.02	103	Hours	Days
5	1352	103	104	Seconds	Days

(d) Pearson correlation

Rank	ID	Rating	Price	Deployment	Leasing
1	7512	-0.38774	102	Seconds	Days
2	1352	-0.38814	104	Seconds	Days
3	1552	-0.38834	105	Seconds	Days
4	9312	-0.38834	105	Seconds	Days
5	1692	-0.38872	107	Seconds	Days

ID 7512 is recommended as the best service because they consider an insignificant increase in the price in contrast to a significant smaller deployment time. Therefore, considering the mapping of these characteristics of a protection service as a vector in space, the Pearson Correlation algorithm is presented as a generally better alternative in contrast to other distance-based similarity algorithms.

A possible alternative to circumvent these differences is given by grouping the vector of protection services for each attribute. Thus, it is possible to compare these service attributes with customer profile attributes in a 1-to-1 manner. Therefore, the final rating of a service is achieved by an average of the rating of its attributes. It should be noted, however, that attributes of protection services offering better conditions than those specified in the customer profile would receive worse ratings. Thus, an alternative can be a rearrangement of input attributes to the best possible conditions, making the recommendation algorithms offer the best alternative possible instead

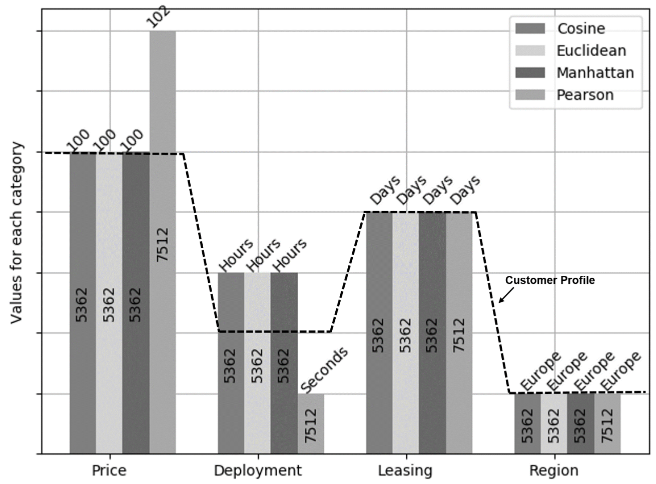


Fig. 5: Best ranked solutions per algorithm in contrast to the customer profile represented by the dotted line

of the closer to the end-user request. For example, if one wants a protection service with deployment time in minutes, protection services a bit more expensive but with deployment time in seconds can be a most suitable recommendation since this still fits the budget and others requirements.

Lastly, such an evaluation indicates that *MENTOR* can recommend adequate protection services considering the price, geolocalization, and other requirements defined by end-users. The distance-based algorithms recommended the cheapest service that is adequate for the end-user according to their demands. However, this service recommended is not necessarily the best one in terms of performance. The Pearson correlation decided toward a bit more expensive service fitting the end-user's budget, while delivering the best performance possible.

A. Discussion and Limitations

Beyond the evaluation concerning the recommendation process provided above, others technical aspects and open challenges are important to be discussed in direction to improve *MENTOR* and also to shed light on further directions for cybersecurity researches on the recommendation of protection services.

Although a large number of protection services are available in the market, this number will arise together with a global deployment of novel paradigms, such as NFV and SDN. Also, novel business models can be used as an incentive for the development of innovative cybersecurity solutions. Based on that, a recommendation system should be able to understand the nuances of services running on different technologies in order to recommend a service efficiently. Besides, mechanisms to deploy the service directly on the customer's infrastructure or in a third-party host should be available, thus simplifying the process of acquisition of such protection services by non-expert end-users.

For the *MENTOR*'s evaluation, 10.000 possible protection services were randomly generated. Such services containing general information (*e.g.*, price, deployment time, and leasing period) helps to demonstrate the feasibility of the solution. However, those services do not represent the real amount of protection services available neither contains exhaustive information of protection services. Most studies should be conducted in order to create a data model (*e.g.*, descriptor) able to define different services and demands, which may include the categorization by technology supported, features provided, and performance aspects.

Also, the reputation of the PSP and protections services itself should be considered during the recommendation process. One should be able to verify the feedback provided by other customers as well as verify performance logs and issues related to past experiences. Besides that, mechanisms to apply penalties to PSP that does not meet the agreement demands should be considered. In such a direction, decentralized reputation mechanisms (*e.g.*, blockchain-based) can be developed to provide a trustworthy and immutable record of reputation regarding the protection services and its different vendors [12].

Another critical aspect of the recommender system is related to the trust of costumers to share data. This discussion is critical, and it is still an open challenge, not only for the *MENTOR* but for other work related to cybersecurity that demands real data to achieve an accurate output. Currently, as a proof-of-concept, it is assumed a consortium of companies and institutions that trust in each one. Thus, one trusted node receives data from customers and offers the *MENTOR* recommendation. Besides, the *MENTOR* is designed to run locally as well, which means that a customer can run his/her own instance of *MENTOR* in a private infrastructure, thus ensuring that the data will not be shared with third-parties.

Lastly, the process of recommending protection services assumes that end-users are able to provide data and the correct parameters to find adequate protection. However, in some cases, users may not know the kind of attack they are under so reactive service matching based on user input would be impractical. In addition, it is a challenge to integrate a recommender system with a variety of logs, because, for example, there is no a single standard of logs when concerning the different type of services and technologies. There is still a lack of mechanisms to deal with the deployment and management of different technologies in an integrated solution, such as APIs and wrappers that help to automate the deployment of recommended services without additional users interactions.

V. Summary, Conclusions, and Future Work

This work introduced *MENTOR*, a protection services recommender system supporting the cybersecurity decision process. The *MENTOR* recommender system maps different customers' requirements to recommend off-site protection services concerning not only price conditions, but also the capacity of services to address specific attacks. In addition, *MENTOR* leverages a competitive market, allowing end-users to acquire services from companies that openly announce their protection services. Also, a modular recommendation engine is provided to support further recommendations algorithms (as openly accessible code [20]). The offering of a dashboard for human interactions in cybersecurity management tasks enables a practical and deployable solution. Since *MENTOR* does additionally offer an open API, the use of such a recommender system within an existing Operation Support System (OSS) can automate decisions to be taken, too.

The mapping of the protection services as well as their attributes enables an accurate evaluation of the similarity between customer requirements and offered security services. *MENTOR*, in this sense, offers a viable approach for the recommendation of services (*e.g.*, possibly offered in open marketplaces based on blockchain). Specifically, the Pearson correlation presented the best balance between cost/benefit considering the mapping of services as a vector. Therefore, in the defined implementation, non-binary characteristics have a significant impact on the evaluation of similarity in contrast to binary ones due to the order of their magnitude, which affects the direction of the vector in space, and as a consequence, its similarity rating.

Although these results are very promising, further investigations are planned in the direction of consolidating the recommendation of protection services, such as by supporting new attributes for the customer profile and services. Also, future work includes: *(i)* investigation of machine learning techniques to combine different similarity measurements, *(ii)* investigation of cybersecurity decisions during real-time cyberattacks, which involves techniques to recognize patterns of different attacks and recommend protections fastly, *(iii)* investigation of recommender systems in the context of service function chaining creation, which involves the determination of which protection services can be part of a chain in order to achieve an adequate level of cybersecurity, and *(iv)* development of a blockchain-based marketplace and reputation system for protection services to introduce a trustworthy public hub, where service providers and independent developers can announce their cybersecurity solutions. Also, technology-agnostic solutions that allows the deployment and management of the different recommended services should be investigated.

Acknowledgements

This work was supported partially by *(a)* the University of Zürich UZH, Switzerland and *(b)* the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement No. 830927, the CONCORDIA project.

Also, the authors would like address many thanks on the insightful discussions to Christian Killer, Eder John Scheid, and Erion Sula.

References

[1] A. Abuhussein, S. Shiva, and F. T. Sheldon, "CSSR: Cloud Services Security Recommender," in *IEEE World Congress on Services (SERVICES)*, San Francisco, USA, August 2016, pp. 48–55.

[2] Akamai, "State of the Internet/Security. DDoS and Application Attacks," 2019, https://bit.ly/2Rr2RRl, last visit August 15, 2019.

[3] X. Amatriain and J. Basilico, "Past, Present, and Future of Recommender Systems: An Industry Perspective," in *10th ACM Conference on Recommender Systems (RecSys 2016*, September 2016.

[4] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, and M. Kallitsis, "Understanding the Mirai Botnet," in *26th USENIX Security Symposium (USENIX 2017))*, Vancouver, Canada, August 2017, pp. 1093–1110.

[5] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutirrez, "Recommender Systems Survey," *Knowledge-Based Systems*, vol. 46, pp. 109–132, 2013.

[6] L. Bondan, M. F. Franco, L. Marcuzzo, G. Venancio, R. L. Santos, R. J. Pfitscher, E. J. Scheid, B. Stiller, F. D. Turck, E. P. Duarte, A. E. Schaeffer-Filho, C. R. P. d. Santos, and L. Z. Granville, "FENDE: Marketplace-Based Distribution, Execution, and Life Cycle Management of VNFs," *IEEE Communications Magazine*, vol. 57, no. 1, pp. 13–19, January 2019.

[7] Cybertango, "The Cybersecurity Directory - DDoS Protection Companies," 2019, https://www.cybertango.io/cybersecurity-vendors/DDoS, last visit August 13, 2019.

[8] B. S. Erion Sula, Muriel Franco, "Design and Prototypical Implementation of Service Recommender System for Distributed Denial-of-Service Attacks," 2019, IfI Bachelor Thesis.

[9] G. Gardikis, K. Tzoulas, K. Tripolitis, A. Bartzas, S. Costicoglou, A. Lioy, B. Gaston, C. Fernandez, C. Davila, A. Litke, N. Papadakis, D. Papadopoulos, A. Pastor, J. Nunez, L. Jacquin, H. Attak, N. Davri, G. Xylouris, M. Kafetzakis, D. Katsianis, I. Neokosmidis, M. Terranova, C. Giustozzi, T. Batista, R. Preto, E. Trouva, Y. Angelopoulos, and A. Kourtis, "SHIELD: A Novel NFV-based Cybersecurity Framework," in *IEEE Conference on Network Softwarization (NetSoft 2017)*, Bologna, Italy, July 2017, pp. 1–6.

[10] M. Husák, J. Komárková, E. Bou-Harb, and P. Čeleda, "Survey of Attack Projection, Prediction, and Forecasting in Cyber Security," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 640–660, September 2018.

[11] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and Other Botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[12] S. Lee, "A Decentralized Reputation System: How Blockchain Can Restore Trust In Online Markets," 2018, https://bit.ly/2w8bpUI, last visit August 12, 2019.

[13] M. Lerato, O. A. Esan, A. Ebunoluwa, S. Ngwira, and T. Zuva, "A Survey of Recommender System Feedback Techniques, Comparison and Evaluation Metrics," in *International Conference on Computing, Communication and Security (ICCCS 2015)*, Pamplemousses, Mauritius, December 2015, pp. 1–4.

[14] T. Li, G. Convertino, R. K. Tayi, and S. Kazerooni, "What Data Should I Protect?: Recommender and Planning Support for Data Security Analysts," in *24th International Conference on Intelligent User Interfaces (IUI 2019)*. Los Angeles, USA: ACM, March 2019, pp. 286–297.

[15] Y. Lu, Z. Zhao, B. Zhang, L. Ma, Y. Huo, and G. Jing, "A Context-Aware Budget-Constrained Targeted Advertising System for Vehicular Networks," *IEEE Access*, vol. 6, pp. 8704–8713, February 2018.

[16] G. Minaev, A. Visa, and R. Pich, "Comprehensive Survey of Similarity Measures for Ranked based Location Fingerprinting Algorithm," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sapporo, Japan, September 2017, pp. 1–4.

[17] S. Morgan, "2019 Official Annual Cybercrime Report," *Herjavec Group*, 2019, https://www.herjavecgroup.com/wp-content/uploads/2018/12/CV-HG-2019-Official-Annual-Cybercrime-Report.pdf, last visit August 15, 2019.

[18] N. Polatidis, E. Pimenidis, M. Pavlidis, and H. Mouratidis, "Recommender Systems Meeting Security: From Product Recommendation to Cyber-Attack Prediction," in *Engineering Applications of Neural Networks*, G. Boracchi, L. Iliadis, C. Jayne, and A. Likas, Eds. Athens, Greece: Springer, August 2017, pp. 508–519.

[19] B. Rodrigues, T. Bocek, A. Lareida, D. Hausheer, S. Rafati, and B. Stiller, "A Blockchain-Based Architecture for Collaborative DDoS Mitigation with Smart Contracts," in *Security of Networks and Services in an All-Connected World, LNCS*, vol. 10356. Zürich, Switzerland: Springer, August 2017, pp. 16–29.

[20] B. Rodrigues and M. Franco, "MENTOR - Protection Services Recommender System," 2019, https://gitlab.ifi.uzh.ch/franco/recommendersystem, last visit August 15, 2019.

[21] P. Rustgi and C. Fung, "DroidNet - An Android Permission Control Recommendation System Based on Crowdsourcing," in *IFIP/IEEE Symposium on Integrated Network and Service Management (IM 2019)*, Washington DC, USA, April 2019, pp. 737–738.

[22] J. Santanna, "DDoSDB: Collecting and Sharing information of DDoS attacks," 2019, https://ddosdb.org/, last visit August 15, 2019.

[23] K. Shah, A. Salunke, S. Dongare, and K. Antala, "Recommender Systems: An Overview of Different Approaches to Recommendations,"

in *International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS 2017)*, Coimbatore, India, March 2017, pp. 1–4.

[24] N. B. Umate and V. G. Bhujade, "A Real Time Technique for Targeted Advertising using Location-based Services For GPS Enabled Device: A Review," in *International Conference of Electronics, Communication and Aerospace Technology (ICECA 2017)*, Coimbatore, India, April 2017, pp. 689–693.

[25] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine Learning and Deep Learning Methods for Cybersecurity," *IEEE Access*, vol. 6, pp. 35 365–35 381, May 2018.

[26] F. Zhang, V. E. Lee, R. Jin, S. Garg, K.-K. R. Choo, M. Maasberg, L. Dong, and C. Cheng, "Privacy-Aware Smart City: A Case Study in Collaborative Filtering Recommender Systems," *Journal of Parallel and Distributed Computing*, vol. 127, pp. 145 – 159, May 2019.

[27] W. Zhang, Y. Wen, and X. Zhang, "Towards Virus Scanning as a Service in Mobile Cloud Computing: Energy-Efficient Dispatching Policy under N-Version Protection," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 1, pp. 122–134, January 2018.

[28] H. Zhu, H. Xiong, Y. Ge, and E. Chen, "Mobile App Recommendations with Security and Privacy Awareness," in *20th International Conference on Knowledge Discovery and Data Mining (SIGKDD 2014)*. New York, USA: ACM, August 2014, pp. 951–960.