

# Quantification label for mobile apps

Sami J. Habib\* and Paulvanna N. Marimuthu

*Department of Computer Engineering, Kuwait University, Kuwait*

*\*Corresponding Author: sami.habib@ku.edu.kw*

**Submitted:** 05/01/2020

**Revised:** 24/12/2020

**Accepted:** 29/12/2020

## ABSTRACT

Nowadays every product we purchase has a tag informing its users how much an automobile or an appliance will cost to run and how much energy a food product will provide on consumption, for example. With the advent of Internet, purchasing Apps becomes habitual, as Apps facilitate easy way of trading and generate money according to App online stores. Currently, Apps provide label that contains a combination of developer info, memory size, category, operating system compatibility, user ratings and reviews, and privacy policy. However, the product label information is asymmetrical and it varies with the developer. It is challenging to develop common standards explicit to users, as their needs are dynamic; moreover, the App undergoes a series of state changes during execution, according to the usage pattern. We propose a quantification label framework to assess the mobile/web App using a set of quantifiers, such as functionality, degree and domain of connectivity, battery consumption, and its vulnerability index for generating a set of standard labels. The corresponding labels in the label frame are: features, popularity, energy consumption and security. Further, we utilized principal component analysis (PCA), a statistical technique to assist the customers in selecting the prominent features from the dataset and to compare related apps, which analyzes the quantifiers in the given datasets and distinguish them according to the variances. The experimental results in quantifying a real time traffic data of an internet based App validated the proposed label by revealing the key features to the users; moreover, PCA carried out on four sets of traffic data enabled the selection of prominent data, and the results also revealed the possible embedding of the framework within an App for dynamic monitoring.

**Keywords:** App Label; Degree of Connectivity; Energy Consumption; Principal Component Analysis; Quantification; Security; Vulnerability.

## 1. INTRODUCTION

The advancement of smart phones and Internet technologies have facilitated Apps to be a ubiquitous and widely used digital space, thereby expediting the development of common to customized Apps for a single application. Today Apps cover almost everything needed for our daily routine, from healthcare, education, business, entertainment, gaming, utilities, social networks and much more. The growth of a number of Apps on a single application makes it difficult for an individual to choose the best one suited for his/her use due to the limited ability to measure the App performance from end-user devices. Moreover, the current information offered by the App developers is asymmetrical, and it is mainly a combination of the following: developer info, platform, version, category, memory size, reviews and ratings, and privacy policy at the time of installation. An efficient market requires information symmetry (Akhawe & Finifter, 2012), as in nutrients labeling of food products (Nayga et al., 1998; Van den Wijngaart, 2002), and the energy labeling in automobiles and appliances (Silitonga et al., 2012), which facilitates the user to select the product of his/her choice. Thus, the mobile Apps should also be equipped with uniform standards/metrics comprehensible to buyers and sellers, so as to enlighten the product quality, security and ease-of-use while executing.

In general, basic software metrics for an App include performance metrics, user and usage metrics, engagement metrics and business metrics. The performance metrics focus on user experience, and the user and usage metrics provide visibility into users' demographics. The engagement metrics deal with the period and frequency of engagement and business metrics highlight the flow on revenue through the App (AppDynamics, 2015). Out of all, the performance metric provides a direct correlation between an application and business, as it provides how fast the App can be installed and how fast it is responding. A fast responsive App may lead to the likelihood of adding large number of users and thus, increasing the revenue. Hence, we have selected performance metrics as the key factor in our quantitative labeling.

The quantitative labeling computes the change in the usage pattern of enterprise/mobile system resources and network components, while executing an App. An App undergoes a series of state changes during its execution: not running, inactive, active, background processing and suspended according to (Dean, 2013). Moreover, the App produces varying traffic loads based on the type of application, say, interdepartmental chat and email applications produce less traffic than a video application. Thus, knowing the maximum size of data packets for defined network connectivity facilitates defining the average energy consumption of the App too, as presented in (Chun & Maniatis, 2009). Thus, it is necessary to monitor the functional connectivity pattern and the usage of system resources to label average response rate of an App under defined bandwidth connectivity and the average memory consumption against usage. Further, security of the App is now given equal importance, since an attacker can steal the credentials and personal data through the installed App. Thus, it will be better if the information, such as the features, responsiveness, power consumption, and level of security of an individual App is accessible to the user before installation while choosing the best one from a pool of Apps.

Alternatively, two or more Apps belonging to a similar category may be compared using principal component analysis (PCA), which is a multivariate classical technique in statistical data analysis, where PCA is mainly utilized for analyzing the variation among different data sets (Zakeri et al., 2014). PCA is applied for prediction, classification, clustering and feature extraction also. It projects the observed data as a linear combination of orthogonal components and finds a direction of lower dimensional space, where the projected data are with maximum total variances (Pei, 2015). We have applied PCA to classify the traffic generated from four Apps to opt out the best performing App.

The main contribution of this work is fourfold. First, we have redefined the App labels based on users' point-of-view rather than developer's point of view. Second, we have generated a state diagram, describing the behavior of an App after installation and we have derived a mathematical model to quantify the state changes during the execution of App. Third, we have derived a relationship between the quantifying numbers of the state changes and the quality of the App to define the constituents of the proposed label frame, explicit to the users. Fourth, we have utilized the principal component analysis (PCA) to compare two similar Apps installed in an enterprise to analyze the impact on the network components.

In this paper, we have refined the labels proposed in our previous work (Habib & Marimuthu, 2019) based on user's point-of-view and we have extended it to develop a quantification labeling to measure the state changes of the App during its execution. The quantification is accomplished by computing the changes in the network components through a set of metrics, which are directly or indirectly influenced by the App. We have refined the parameters, such as degree of connectivity, and domain of connectivity from our prior work (Habib et al., 2018) to reflect network usage in the existing states, and we have further added new metrics, such as energy consumption and vulnerability to strengthen the label quality. Then, the quantified metrics are applied to generate the label constituents, such as, features, popularity, security, and energy consumption so as to structure the label frame. A principle component analysis module is included in the labeling to analyze the most impacted network parameter and to compare the performance of four similar Apps based on changes in network components. The simulation results on testing the activity of a real App demonstrated the feasibility of framing common App standards and the possible extension to embed the code as a construct within the App for dynamic monitoring.

The rest of the paper is organized into six sections. Section 2 details the related work, and Section 3 describes the various states of an App and the metrics to quantify the state changes. Section 4 details the principal component analysis, and Section 5 details the proposed labeling and the proposed structure of an App label frame. Section 6 presents the outcomes of the experiments and Section 7 concludes the paper.

## 2. RELATED WORK

Our literature survey on labeling on software App reveals that there are no common standards developed yet and we would like to take this opportunity to provide the readers about the importance of uniform labeling, as there are many low-quality, unsafe and even potentially harmful Apps causing more risks to the public (Lewis, & Wyatt, 2014; Huckvale et al., 2015). However, the assessment of App is still challenging problem and many of the current research works followed customer reviews and ratings to evaluate them. Studies on online market Apps, such as online book, groceries and clothes stores have analyzed the App based on customer review and ratings (Mojica Ruiz et al., 2016; Chevalier & Mayzlin, 2006; Harman et al., 2012). Akhawe and Finifter (2012) briefed about the future of adding labels to mobile Apps, and they mainly narrowed down their discussion of adding security information to the Apps. The authors discussed the necessity to label the Apps but they did not either mention any specific approach or perform any experiment. Jebari and Wani (2012) proposed a centroid based approach to classify web pages by genre, where they utilized information sources such as URL, title and anchors to extract the details. Sun et al. (2017) discussed to add labels to know the type of user's psychological needs for the App satisfaction. In this work, the authors concentrated on in-depth interviews and review data samples to derive the psychological needs of app users. There is no quantification of app usage statistics to standardize the app labels. Chen et al. (2016) proposed a labeling to tag the mobile App with few keywords for improving the search, categories and advertising.

Furthermore, the literature survey on quantitative and qualitative analyses of Apps reveal that a few works utilized quantitative and qualitative study in analyzing App performance. A quantitative study of educational mobile Apps was carried out by Hayes (2016) to analyze few parameters, such as the grade level, subject, learning ratings and skill category. Here, the analysis was limited by the nature of data collected and view of the individuals on the selected parameters. Jiang et al. (2018) utilized a new mobile application crawler to gather information, and they conducted a quantitative analysis on the collected data to study the average latency of mobile applications. Alderson et al. (2005) proposed a qualitative approach to model a real-time internet traffic, where they analyzed the degree of connectivity against the bandwidth at a router level. The authors carried out a coarse-grain approach to study the connectivity variations at the physical layer.

Few researchers analyzed App metrics, such as energy consumption and security individually. Xiao et al (2014) discussed the energy consumption characteristics of an App using Wi-Fi communication, while Cho et al. (2013) presented a vulnerability analysis on data sharing under android platform. Diez et al. (2017) carried out an audit of security of an mHealth Android application, where they attempted to decompile the application. Chen et al. (2020) proposed a three-phase automated security risk assessment system, which utilized a combination of statistical analysis and keyword identification techniques to categorize the weakness in the existing mobile bank app. The authors claimed that they were able to list out 2157 weaknesses in 693 bank Apps. Alotaibi et al. (2020) discussed the energy issues by the mobile App and the use of existing classification techniques in categorizing the issues.

None of the previous works focused on designing standardized labeling of Apps. In this paper, we have made a novel attempt to label the Apps so as to reveal App constituents to the users.

## 3. QUANTIFYING THE STATE CHANGES

When an App is executing, it undergoes five processing states namely: not running (NR), inactive (IA), active (AC), background processing (BG) and suspended (SP). The state diagram of an App process is illustrated in Figure 1, where the App in its *not running* (NR) state after the installation, and then switches to *inactive* (IA) state so as to receive calls. Once it is requested, then moves to *active* (AC) state. At task completion or receiving an interrupt, it returns to IA. Then, it runs in *background* (BG), if it is suspended temporarily on receiving a second request simultaneously and then it switches to (AC) state through (IA) state to run in foreground. If the device power off, it enters to (BG) state from (AC) state. The process enters into *suspended* state (SP) upon receiving a forced quit while in BG or AC state. Then the process is terminated and the App process enters into *NR* state.

We have selected parameters, such as degree of connectivity, domain of connectivity, energy consumption during data transmission, and vulnerability, to quantify the state changes during App execution, and the parameter selection

is made compatible to reflect the number of users connected in various domains, say  $k = \{d_1, d_2, \dots, d_m\}$  in various processing states, during App execution.

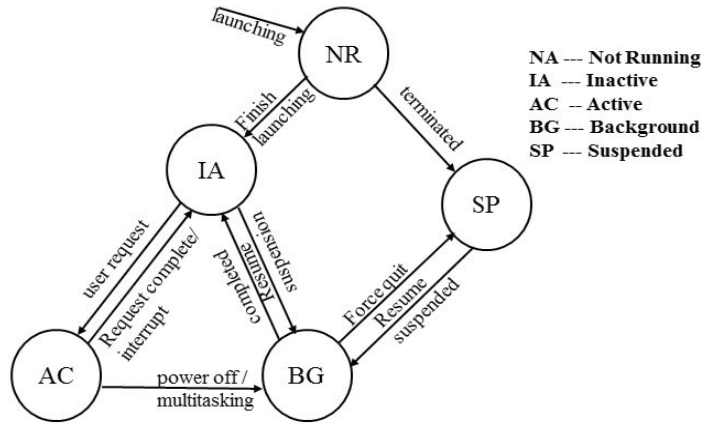


Figure 1. State diagram of an app process.

### 3.1 Degree of Connectivity

We have defined the *degree of connectivity* of an App as the maximum connectivity established between the App (target node) and other spatially separated users in the active state ( $AC \in \alpha$ ), as illustrated as in Equation (1), where the term  $\alpha$  specifies the set of processing states. The events are received during the active state only; however, the connectivity is retained in (BG) state. Here, we have considered the active state only. Let  $d_{ij}$  be the maximum degree of the testing App  $n_{ij}$  ( $n_i$  is the users’ device where the App is going to be installed) in a domain  $j$  in the active state. The domain  $j$  takes on three values  $\{1,2,3\}$  to represent the connected the users, namely local, extended, and global respectively.

$$d(n_i)|_{\alpha}^k = \sum_{j=1}^k n_{ij} \tag{1}$$

### 3.2 Domain of Connectivity

The domain of connectivity is defined as the sum of degree of users ( $w_{jk}$ ) using the  $App_j$  across various domains  $K = \{k_1, k_2, \dots, k_m\}$  while it is in a state  $\alpha_i \in \alpha$ , shown in Equation (2). Here, the numerator specifies the degree of connectivity of the App with users distributed in  $m$  domains in state (AC). The term  $w_k$  represents the total number of connected users, using various Apps in  $k$  domains, and the term  $k = 1, 2, \dots, m$ .

$$DC(App_j)|_{AC} = \sum_{i=1}^N \sum_{k=1}^m \frac{w_{i,k}^{App_j}}{\sum_{k=1}^m w_k} \quad AC \in \alpha \tag{2}$$

### 3.3 Energy Consumption

The growth of applications on wireless platform poses a threat to battery lifespan, as the high states execution within applications accounts for higher energy consumptions (Venkataraman & Muntean, 2012). The App consumes power based on the hardware usability index and its duration of connectivity with other users. Thus, we have defined the energy consumption of an App as the sum of average energy consumption in all its processing states and its state transitions, as illustrated in Equation (3). Here, the term  $\sigma_{i,q}$  is a Boolean variable stating the occurrence of state changes during the specified timespan  $t$ . The App is connected externally to send or receive data in active state, which

is defined in terms of number of packets transmitted during the connected timespan as defined in Equations (4) and (5). Here the parameter  $N$  denotes the maximum number of connected users, and the term  $C_{n_i, n_j}$  is a Boolean variable, which is true if there is a connection between  $n_i$  and any user  $n_j$ . The parameter  $e$  is the energy consumed during data transmission. Here, the application type limits the size of the file to be transmitted and hence, the energy of a packet transmission is fixed for an application.

$$E(n_i)|_t = \sum_{q=1}^{\alpha} E_q(n_i)|_t + \sum_{\substack{i,q \\ i \neq q}}^{\alpha} E_{i,q}(n_i)|_t * \sigma_{i,q} \quad \text{where, } \sigma_{i,q} \in \{0,1\} \quad (3)$$

$$E(n_i)|_t^{\alpha_i} = \sum_{n_i \in N} \sum_{n_j \in N} C_{n_i, n_j} * (tr(l) * e) \quad \text{where } \alpha_i \in \{AC\} \subset \alpha \quad (4)$$

$$tr(l) = \text{No. of transmissions} = \frac{\text{Size of available data}}{\text{Maximum packet size permitted by the App}} \quad (5)$$

The energy consumption in states, such as not running, inactive, and suspended is negligible compared to active state.

### 3.4 Security

We have considered an App with finite connectivity indexed by  $N=1,2,3,\dots,q$  and with  $\alpha_i \in \alpha$  denoting the state of the App, we rate the security of the App from the evaluation of its vulnerability index while executing App, as shown in Equation (6). We have defined the vulnerability index as a sum of relative measure of factors leveraging threat to its security, such as remote access (RA), surge in memory usage (MU), and Battery usage (BU) as in Equation (7).

$$Se(n_i)|_t^{\alpha} = 1 - V(n_i)|_t^{\alpha} \quad (6)$$

$$\text{where, } V(n_i)|_t^{\alpha} = V(n_i)|_{RA}^{\alpha} + V(n_i)|_{MU}^{\alpha} + V(n_i)|_{BU}^{\alpha} \quad (7)$$

The remote access vulnerability (RA) is estimated using closeness centrality, which is defined as in Equation (8), as a ratio of degree of connectivity (undirected graph) of the App from users in  $k$  domains to the sum of shortest paths between the App and connected users ( $ds(App, cu_j^k)$ ). The closeness centrality ( $cc$ ) measures the App connectivity distribution in a given timespan  $t$ , and it ranges from zero to one. Here, the term  $ds$  refers to the shortest distance, the term ( $App$ ) is the subject under evaluation and the term  $cu_j^k$  refers to the total connected users in  $k$  domains. The App with a larger closeness centrality may be more secure to targeted attacks. We have estimated the vulnerability index in active (AC) and background states (BG), as they are communicating externally.

$$V(n_i)|_{RA}^{\alpha_i} = \frac{\sum d_i((App), m)}{\sum_{k=1}^m \sum_{j=1}^N ds(App, cu_j^k)} \quad \text{where, } \alpha_i = \{AC, BG\} \subset \alpha \quad (8)$$

The vulnerability due to surge in memory consumption in any states is defined as in Equation (9), where the expected value of memory usage and tolerance for a given App are given by the developer in the label frame. If the estimated memory consumption is deviating very much from the tolerance, then the risk is higher.

$$V(n_i)|_{MU}^{\alpha} = \frac{\left( \frac{\sum MU(n_i) - E[MU(n_i)]}{\alpha} \right)}{E[MU(n_i)]} \quad \text{where, } \alpha_i = \{AC, BG\} \subset \alpha \quad (9)$$

The surge in battery consumption is one of the high risk factors to security, since a sudden drain of battery may occur due to illegal accessing of internal data or sending credentials to a remote node. The percentage of deviation from the expected value as given in Equation (10) will give a notion of threat to user.

$$V(n_i)|_{BU}^{\alpha_i} = 1 - \frac{\left( \frac{\sum BU(n_i) - E[BU(n_i)]}{\alpha_i} \right)}{E[BU(n_i)]} \text{ where, } \alpha_i = \{AC, BG\} \subset \alpha \quad (10)$$

Finally, we have formulated a weighted linear equation of the three defined vulnerability factors, as demonstrated in Equation (11), where the weight factor is selected based on the sensitivity of the application.

$$V(n_i) = \eta_1 * RA(n_i) + \eta_2 * BU(n_i) + \eta_3 * MU(n_i) \text{ where, } \eta_1 + \eta_2 + \eta_3 = 1 \text{ and } \left. \begin{matrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{matrix} \right\} \geq 0 \quad (11)$$

### 4. PRINCIPAL COMPONENT ANALYSIS

The principal component analysis is a statistical technique, which is mainly applied to find patterns in a dataset, and to select the most significant feature producing high impact among the given features in the dataset. It provides mapping of a dataset with few variables into a new dataset with less uncorrelated variables, each of which is represented as a linear combination of the original variables, known as the principal components. The principal components are estimated from the Eigenvectors of the covariance matrix of the original variables and they are ordered by the amount of variation in the given data. The first principal component captures the largest variance and it is represented by drawing an axis in the direction of maximum variation of the given data. A second axis is added in the direction orthogonal to the first to display the next highest variation, which is the second principal component; then, the subsequent principal components capture the residual variations sequentially in a descending order.

#### 4.1 Comparing Traffic Data Sets using PCA

Let  $X$  be a table of traffic datasets observed from a number of Apps, where each dataset has  $i$  observations, described by  $j$  variables (features) and each table is represented by a matrix  $X_{i,j}$  having  $i$  rows and  $j$  columns. As a first step, we have standardized the given data matrix  $X_{i,j}$  with zero mean and unit variance, since the variables in the dataset are measured on different scales and have different units. Then, a covariance matrix of the given data matrix  $X_{i,j}$  is constructed with size  $(j \times j)$ , where each row represents the dependence among the other selected variables. Here, for the given matrix of size  $(i \times j)$ , having  $i$  observations and  $j$  variables, where  $i=5$ , and  $j=3(a, b, \text{ and } c)$ , its covariance matrix is obtained as a matrix of size  $(3 \times 3)$  as shown in Equation (12) and it takes three values: positive, negative and zero. A positive covariance indicates that both the features are changing together, whereas a negative covariance indicates a converse relationship and a zero demonstrates no relation.

$$\begin{bmatrix} \text{cov}(a,a) & \text{cov}(a,b) & \text{cov}(a,c) \\ \text{cov}(b,a) & \text{cov}(b,b) & \text{cov}(b,c) \\ \text{cov}(c,a) & \text{cov}(c,b) & \text{cov}(c,c) \end{bmatrix} \quad (12)$$

Subsequently, we have estimated the Eigenvectors and Eigenvalues from the covariance matrix, where the Eigenvectors of the principal component represent the direction of new features space and the Eigenvalues represent the magnitudes. Then, a projection matrix is constructed with the selected  $r$  features from  $j$  features, such that  $r < j$ , where  $j$  is the original data matrix variables; the  $r$  features are selected after sorting the computed Eigen values in descending order. This facilitates the transformation of original data matrix through the projection matrix to obtain the new R-dimensional feature subspace.

### 5. PROPOSED METHODOLOGY TO DERIVE FRAMEWORK FOR APP LABEL

The proposed labeling may be embedded as a code construct within the App during App development to update the defined parameters periodically over a specified timespan  $t$ . The pseudocode construct is shown in Figure 2, and it is continuously recording the defined App parameters and exploring the data to update its popularity, energy consumption and memory consumption and maximum responsiveness to the users.

We have proposed a structure of App label frame as illustrated in Figure 3, where we have defined the labels, such as features, popularity, energy consumption, and security. The features detail the functions of the App, and it should be given by the developer to know how they match with users’ requirements. The popularity is classified into three, based on the range of connected domains: i) local, if the users are from the local hub (domain), ii) extended, if the users are connected widely and iii) global, if connectivity is across the borders and the range is high. Then, the second label, energy consumption, details the average energy consumed over the stipulated timespan during the execution and the third label, security, is defined in a scale from zero to one. The App is unsecure, when the value is close to zero and its moderate if it is around the midpoint and highly secure when it is close to one.

```

Dynamic Code Construct for App_Label ()
{
Generate maximum connectivity pattern for the App();
Observe and log parameters (node_deg, traffic, domain, battery_use, and memory_use);
Set thresholds of battery_use and memory_use parameters from_observations ();
Estimate_popularity over defined timespan t ();
Estimate_security of App ();
Update_observation_period () (t = t+ Δt);
Update_App_labels ();
}
    
```

**Figure 2.** Pseudocode construct for App labels generation.

Nutrition Facts of App	
Installation date	
Features	functions
Popularity	local
	Extended
	High
Energy consumed	X Joules
Security	<div style="display: flex; justify-content: space-between;"> <span>0</span> <span>1</span> </div> <div style="display: flex; justify-content: space-between;"> <span>low</span> <span>high</span> </div>

**Figure 3.** App label frame.

## 6. RESULTS AND DISCUSSION

Our experiment test scenario is comprised of sample data comprising traffic pattern recorded for 30 days period (Caida, 2019), which is added with features, such as the node degree, connectivity, domain information to reflect an App behavior. We have carried out a set of experiments to estimate the parameters pertaining to initial phase of labeling the App.

### 6.1 Popularity

The node degree of connected users in the selected range, as explained in section 5, the frequency of connectivity, and the duration of connectivity ( $t$ ) are the three parameters utilized to conclude the popularity of a given App, and the popularity of an App is defined as in Equation (13). The popularity of App in various domains is demonstrated in Figure 4, where each section represents range of connectivity and the number of connections. Here, there are more numbers of connectivity observed in higher connectivity range (70%), thereby indicating high popularity.

$$P = \sum (\text{deg}(\text{App}) * t * \text{domain}) \quad (13)$$

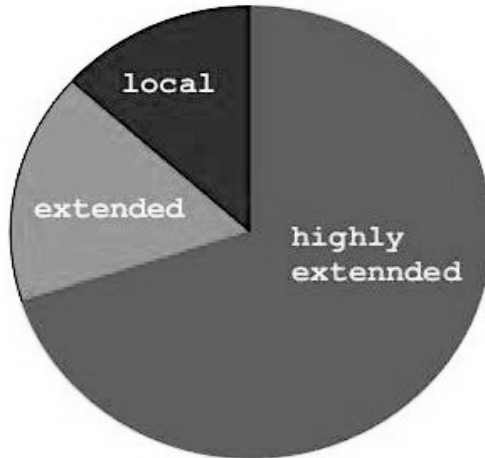


Figure 4. Popularity of the selected App.

### 6.2 Energy Consumption

We carried out an energy calculation for the selected App traffic, where we made few assumptions. In the experimental setup, we have considered a fixed data rate of 1500 kbps and the average power consumed during the selected data rate as 300 mill watt, according to (Caida, 2019). We have considered a fixed packet size of 1472 bytes, according to (Fall & Stevens, 2011) and we have estimated the duration of each packet and the total number of packets generated for a specific traffic. The battery capacity is taken as 1369 mAh, rated at 5V and the corresponding energy is 24642 J. We estimated the energy consumption against the number of packets and its ratio against the given maximum energy limit during the App execution. The energy consumption specific to day traffic is found to be varied between 14 J and 39 J, which exceeds the upper limit of 35 J in 10 percent (3 days /30 days) of the total observations.

### 6.3 Vulnerability Index

We have utilized three parameters, that is, closeness centrality, battery consumption and memory consumption to estimate the vulnerability index of a data transfer. Equation (11) in Section 3.4 is utilized to compute the vulnerability index, where the parameters  $\eta_1$ ,  $\eta_2$ , and  $\eta_3$  are selected as 0.4, 0.3 and 0.3 respectively to give importance to the domain of connectivity. The maximum node degree for each data transaction is restricted to 50 and a distance matrix detailing the node distances corresponding to all connectivity is included. In this experiment scenario, the closeness



centrality is given more weightage, as the sparsely distributed nodes may be prone to personal attack easily. In each category, the outcomes of the experiment are assigned with lower and upper threshold values and three ranges are derived as in Equation (14) based on the selected threshold values. Figures 5 and 6 demonstrate the observed data and its vulnerability index. It is observed from Figure 6 that 6% of the observed data falls near the upper limit of the 0 to 1 scale of vulnerability index and 23% data fall near the lower limit and the rest 71% is in the normal range, thus showing the risk index is low and security is high.

$$\begin{aligned}
 \text{range\_1} &= 0 < \text{range1} < \text{lower} \\
 \text{range\_2} &= \text{lower} \leq \text{range2} < \text{upper} \\
 \text{range\_3} &= \geq \text{upper}
 \end{aligned}
 \tag{14}$$

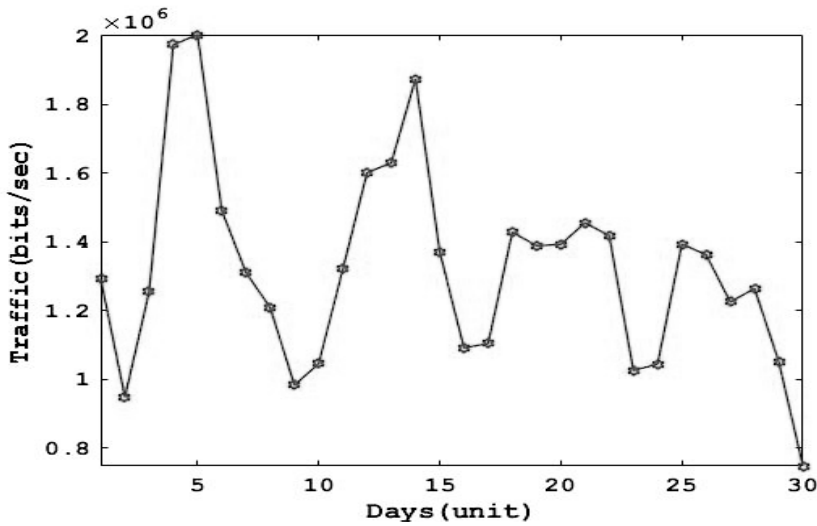


Figure 5. App traffic pattern.

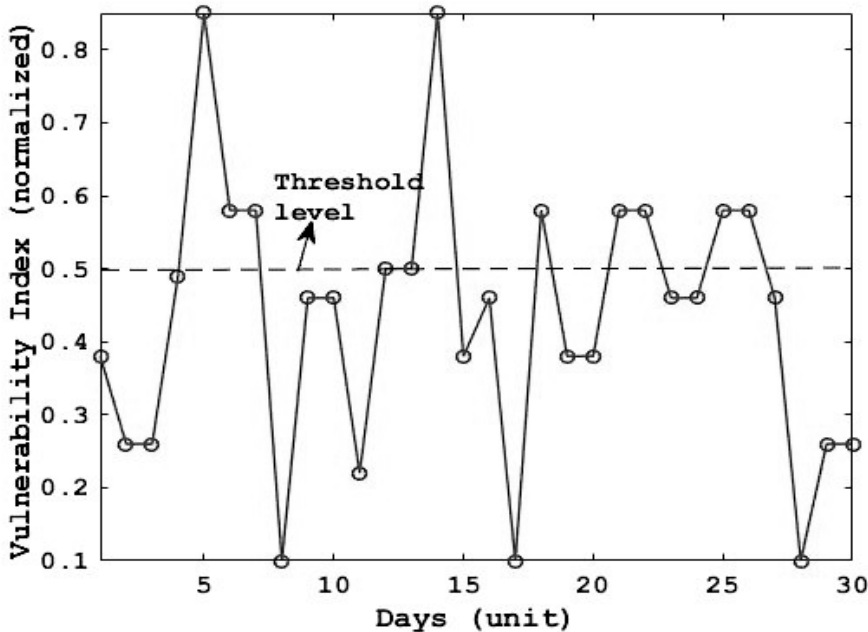


Figure 6. Vulnerability Index of traffic pattern in Figure 5.

Finally, we draw the App label frame for the above selected application, as in Figure 7, which summarizes the statistical constituents of the App from user point of view.

Nutrition Facts of App	
Installation date	
Features	ecommerce
Popularity	High
Energy consumed	14 to 34 Joules
Security	

Figure 7. App Label Frame.

#### 6.4 Validation of Proposed Framework

We carried out a second set of experiments on a real-time traffic dataset (kaggle, 2017) to validate the proposed framework, where the traffic dataset is comprised of frequency of transactions, amount of data transfer, duration of connection, number of forwarded packets, number of backward packets, length of packets, and energy usage for various Apps. We extracted the data of three Apps for 100 observations. We applied PCA to select the prominent features out of seven given parameters in the dataset. The prominent features, showing high variances are: traffic flow and connection frequency. We compared the selected Apps with respect to the traffic flow and the resultant covariance and scores of PCA analysis are presented as a three dimensional graph using biplot, as shown in Figure 8. The graph is used to distinguish the prominent features in a particular run. The parameters component 1, component 2 and component 3 represent the first three principal components of the dataset; where the lines labelled by App-1 to App-3 represent the correlations of Apps with each other. App-1 and App-2 are closely related, as those Apps showed positive correlation and an angle deviation less than 90°. App-3 showed a large angle deviation, thus demonstrating negative correlation with App-1 and App-2. The scores (dots in the graph) on each row demonstrate the positive or negative impact of selected features on the given App.

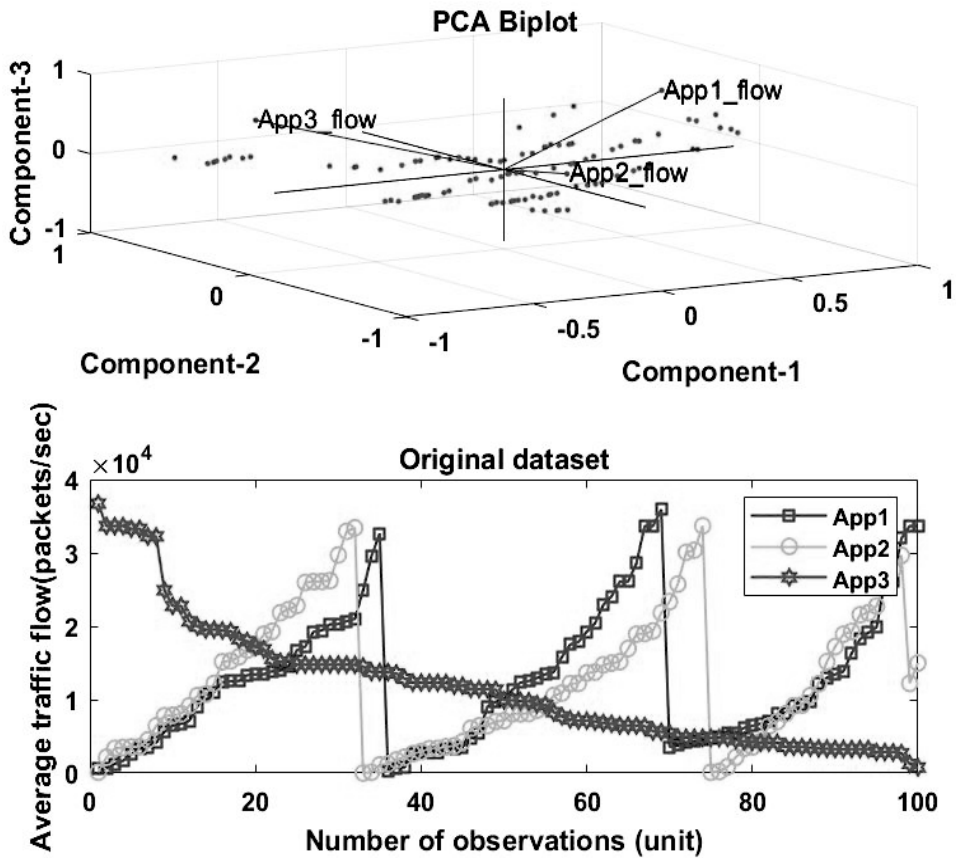


Figure 8. Principle component analysis.

The second plot in Figure 8, demonstrated the behavior of original dataset. On comparing the biplot with the original dataset exhibiting the Apps behavior, it is observed that App-3 showed a negative trend which is very well matched from the biplot conclusions.

Further, we computed the variability index of the Apps, as presented in Table 1. The variability index showed that the App, which is showing highest variation is contributing more in the assessment of the selected parameters, in this case, the traffic flow. This might be due to increase in the popularity and more usage of the App.

Table 1. Variability Index of the Apps with respect to traffic flow.

App	Variations
App-1	59.67
App-2	22.95
App-3	17.37

The connection frequency is presented in Figure 9. It is observed that App1 is highly connected and it is popular among the three. It also showed higher variations during data transfer, as tabulated in Table 1.

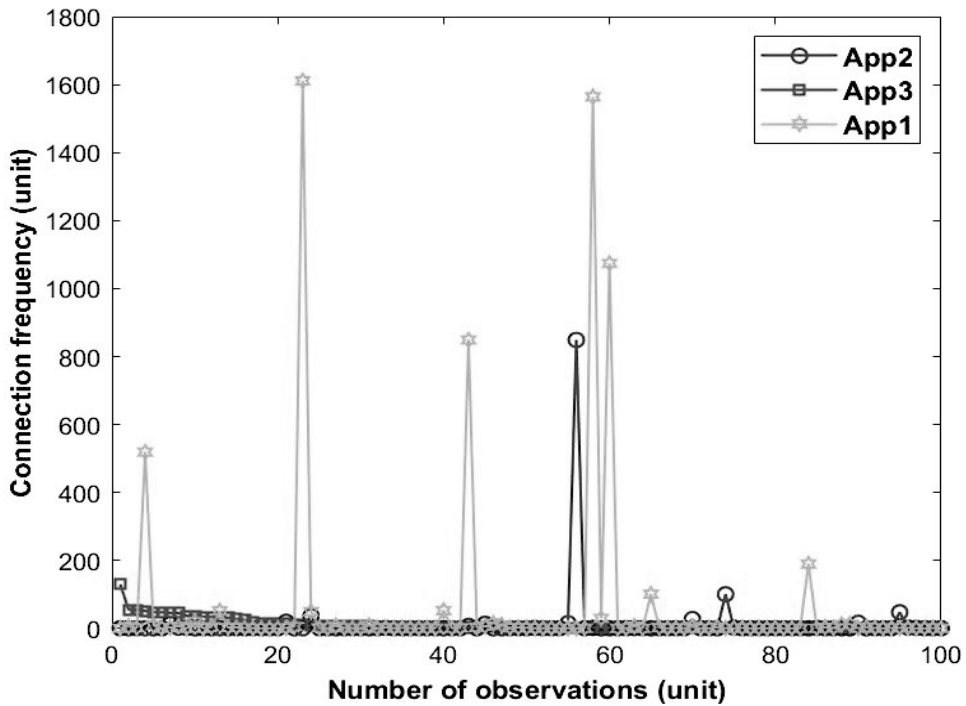


Figure 9. Connection frequency variations observed among the Apps.

We estimated the energy consumed by the three Apps, where we utilized the parameters, such as, frequency of connection, duration of each connections, and the number of packets transferred to estimate the energy. We assumed  $k$  units of energy consumed by each packet transfer, and the average energy transfer for 100 observations is shown in Figure 10. App1 consumed 12% more than App2 and 35% more than App3. This demonstrated that App3 is not popular and with lower transactions.

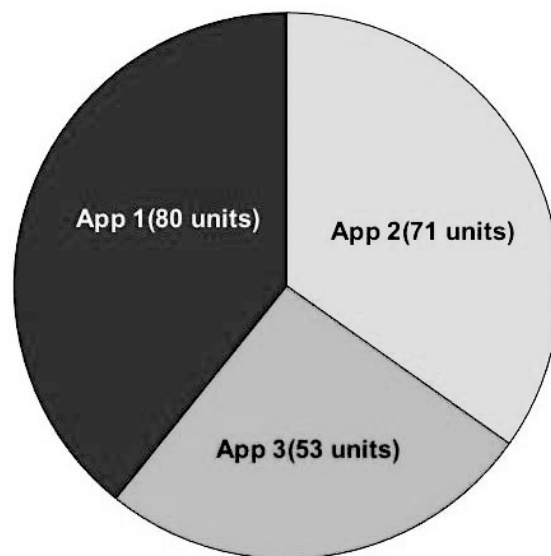
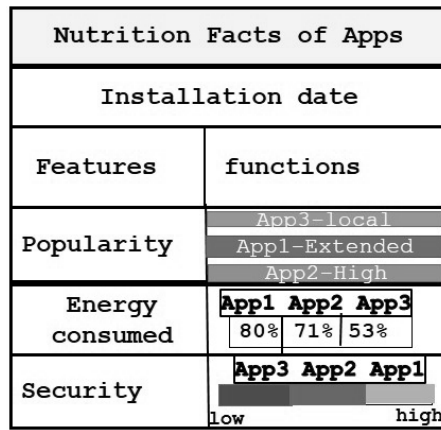


Figure 10. Energy consumption observed among the Apps.

The security level is computed by analyzing the variance in the connectivity, which is listed in Table 2. More variations (90%) are observed in App1 compared to others, while App2 and App3 contributed only 10% variations. On consolidating the connection frequency and energy parameters, App1 tops the security. The summary of App labels for the selected comparison study is presented in Figure 11, which guides the users in choosing right App according to their requirements.

**Table 2.** Variability Index of the Apps with respect to connection frequency.

App	Variations
App-1	90.06
App-2	9.49
App-3	0.44



**Figure 11.** Comparison of App labels.

## 7. CONCLUSION

We have proposed a quantification labeling to assess mobile/web App’s quality to facilitate the users to know the features, responsiveness, security, and energy consumption before selecting an App from a group of similar Apps in the market. Our labeling accounted the series of states the App undergoes during its execution by quantifying the changes in the mobile network components. The degree and domain of connectivity, energy consumption and vulnerability were the set of parameters selected to quantify the state changes, and subsequently, utilized to form an App label frame, which is comprised of four labels: features, popularity, energy consumption, and security. Further, the labeling applied principal component analysis, a statistical technique, to compare the traffic pattern of two Apps running simultaneously to categorize the prominent one impacting the network parameters. The experimental results in quantifying a real time traffic data validated the proposed framework by deriving information necessary to form the labels. The principal component analysis carried out on two sets of traffic data enables the selection of the prominent App features, which revealed the possibility of embedding of the framework within an App for dynamic monitoring. We validated the proposed framework using a real dataset, and the estimated quantification parameters revealed the suitability of the proposed framework for framing App labels.

## ACKNOWLEDGMENT

This work was supported by Kuwait University under a research grant no. QE01/17

## REFERENCES

- Akhawe, D., and Finifter, M. 2012.** Product labels for mobile application markets. In the Proceedings of Mobile Security Technologies Workshop, 24 May, San Francisco, CA, USA.
- Alderson, D., Li L., Willinger W., and Doyle J. 2005.** Understanding internet topology: principles, models, and validation. *IEEE/ACM Transactions on Networking*. vol. 13, no. 6, pp. 1205-1218.
- Alotaibi, A., Clause, J., and Halfond, W.G.J. 2020.** Mobile App Energy Consumption: A Study of Known Energy Issues in Mobile Applications and their Classification Schemes – Summary Plan, In the Proceedings of IEEE International Conference on Software Maintenance and Evolution, Adelaide, Australia, pp. 854-854.
- Chevalier, J.A, and Mayzlin, D. 2006.** The effect of word of mouth on sales: online book reviews, *J. Marketing Research*, vol. 43, no. 3, pp. 345-354.
- Chun, B.G., and Maniatis, P. 2009.** Augmented smart phone applications through clone cloud execution, In the Proceedings of the 12th Workshop on Hot Topics in Operating Systems, USENIX Association, May 18-20, Monte Verita, Switzerland, pp. 1–8.
- Cho, T., Kim, J-H., Cho, H-J., Seo, S-H., and Kim, S. 2013.** Vulnerabilities of android data sharing and malicious application to leaking private information, In the Proceedings of Fifth International Conference on Ubiquitous and Future Networks, 2-5 July, Da Nang, Vietnam, pp. 37-42.
- Chen N., Hoi S.C.H., Li S., and Xiao X. 2016.** Mobile app tagging. In the Proceedings of 9th ACM international conference on web search and data mining, 22 – 25 February, San Francisco, California, USA, pp. 63-72.
- Chen, S, Fan, L., Meng, G., Su, T, Xue, M., Xue, Y., Liu, Y., and Xu, L. 2020.** An Empirical Assessment of Security Risks of Global Android Banking Apps, In the Proceedings of IEEE/ACM 42nd International Conference on Software Engineering (ICSE), Seoul, pp. 1310-1322.
- Dean, G. 2013.** Understand the states and transitions of an iOS App. Downloaded from <https://www.techrepublic.com> on 12-September 2019.
- Fall K.R and Stevens W.R. 2011.** TCP/IP Illustrated-volume I. Addison Wesley, Michigan, USA.
- Harman, M., Jia, Y., and Zhang, Y. 2012.** App store mining and analysis: MSR for App stores, In the Proceedings of 9th IEEE Working Conf. Mining Software Repositories, June 2-3, Zurich, Switzerland.
- Hayes, T. 2016.** Mobile Apps for 21st Century Skills: A quantitative analysis of educational mobile apps on graphite.org. In Proceedings of 2016 World Conference on Educational Media and Technology. 28-30 June, Vancouver, BC, Canada, pp. 1630-1637.
- Habib S.J., Marimuthu P.N., and Rajasundari, T. 2018.** Quantification of new web applications within enterprise networks. In the Proceedings of International Conference on Information Integration and Web-based Applications and Services, 19-21 November, Yogyakarta, Indonesia.
- Habib S.J., Marimuthu P.N. 2019.** App nutrition label. In the Proceedings of World Conference on Information Systems and Technologies, 16-19 April, Galicia, Spain.
- Huckvale, K., Prieto, J.T., Tilney, M., Benghozi, P.J., and Car, J. 2015.** Unaddressed privacy risks in accredited health and wellness Apps: A cross-sectional systematic assessment. *BMC medicine*, vol. 13, no. 214, pp.1-13.
- I. de la Torre-Díez, B.O. Trinchet, J.J.P.C. Rodrigues and M. López-Coronado. 2017.** Security analysis of a mHealth app in Android: Problems and solutions, In the Proceedings of IEEE 19th International Conference on e-Health Networking, Applications and Services, Dalian, China, pp. 1-6.
- Internet traffic traces, [https://www. Caida.org/data](https://www.Caida.org/data) accessed on September 2019.
- Jebari C., and Wani, M.A. 2012.** A multi-label and adaptive genre classification of web pages. In the Proceedings of 11th International Conference on Machine Learning and Applications, December 12-15, Boca Raton, FL, USA, pp. 578-581.
- Jiang Z., Kuang R., Gong J., Yin, H., Lyu, Y. and Zhang, X. 2018.** What makes a great mobile app? A quantitative study using a new mobile crawler. In the Proceedings of IEEE Symposium on Service-Oriented System Engineering, 26-29 March, Bamberg, Germany, pp. 222-227.

- Lewis, T.L., and Wyatt, J.C. 2014.** MHealth and mobile medical apps: A framework to assess risk and promote safer use, *Journal of Medical Internet Research*, vol. 16, no. 9. pp. 1-10.
- Mojica Ruiz, I.J, Nagappan, M., Adams, B., Berger, T., Dienst, S., and Hassan, A.E. 2016.** Examining the rating system used in mobile-App stores, *IEEE Software*, vol. 33, no. 6, pp. 86-92.
- Nayga R.M, Lipinski D, and Savur N. 1998.** Consumers' use of nutritional labels while food shopping and at home. *Journal of Consumer Affairs*, vol. 32, no. 1, pp. 106-120.
- Pei, Y. 2015.** Linear principal component discriminant analysis. In the Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, October 9-12, Hong Kong, pp. 2108-2113.
- Silitonga A.S., Atabani A.E., and Mahlia, T.M.I. 2012.** Review on fuel economy standard and label for vehicle in selected ASEAN countries, *Renewable and Sustainable Energy Reviews*, vol. 16, no. 3, pp. 1683-1695.
- Sun Z., Ji Z., Zhang P., Chen C., Qian X., Du X., and Wan Q. 2017.** Automatic labeling of mobile apps by the type of psychological needs they satisfy. *Telematics and Informatics*, vol. 34, no. 5, pp. 767-778.
- Traffic dataset: <https://www.kaggle.com/jsrojas/ip-network-traffic-flows-labeled-with-87-apps>, accessed on 12 December 2020.
- Van den Wijngaart A.W. 2002.** Nutrition labelling: purpose, scientific issues and challenges, *Asia Pacific Channel of Clinical Nutrition*, vol. 11, no. 2, pp. 68 -71.
- Venkataraman, H., and Muntean G-M. 2012.** Green mobile devices and networks energy optimization and scavenging techniques. CRC Press. NY, USA
- Xiao, Y., Cui, Y., Savolainen, P., Siekkinen, M., Wang, A., Yang, L., Yla-Jaaski, A., and Tarkoma, S. 2014.** Modeling energy consumption of data transmission over Wi-Fi, *IEEE Transactions on Mobile Computing*, vol. 13, no. 8, pp. 1760-1773.
- Zakeri, V., Tavakolian, K., Arzanpour, S., Zanetti, J.M., Dumont, G.A., and Akhbardeh, A. 2014.** Preliminary results on quantification of seismo-cardiogram morphological changes, using principal component analysis. In the Proceedings of 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, August 26-30, Chicago, IL, USA, pp. 6092-6095.
- 16 Metrics to Ensure Mobile App Success.** Downloaded on 10-September-2019 <https://www.appdynamics.com/media/uploaded-files/1432066155/white-paper-16-metrics-every-mobile-team-should-monitor.pdf>