symmetry

MDPI

Article

# ContextPCA: Predicting Context-Aware Smartphone Apps Usage Based On Machine Learning Techniques

**Iqbal H. Sarker [1,2,*], Yoosef B. Abushark [3] and Asif Irshad Khan [3]**

[1] Department of Computer Science and Software Engineering, Swinburne University of Technology, Melbourne VIC-3122, Australia
[2] Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chittagong 4349, Bangladesh
[3] Computer Science Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; yabusharkh@kau.edu.sa (Y.B.A.); aikhan@kau.edu.sa (A.I.K.)
* Correspondence: msarker@swin.edu.au

Received: 4 February 2020; Accepted: 16 March 2020; Published: 1 April 2020


**Abstract:** This paper mainly formulates the problem of predicting context-aware smartphone *apps usage* based on machine learning techniques. In the real world, people use various kinds of smartphone apps differently in different contexts that include both the *user-centric* context and *device-centric* context. In the area of artificial intelligence and machine learning, *decision tree* model is one of the most popular approaches for predicting context-aware smartphone usage. However, real-life smartphone apps usage data may contain *higher dimensions of contexts*, which may cause several issues such as increases model *complexity*, may arise *over-fitting* problem, and consequently decreases the *prediction accuracy* of the context-aware model. In order to address these issues, in this paper, we present an effective *principal component analysis (PCA)* based context-aware smartphone apps prediction model, *"ContextPCA"* using decision tree machine learning classification technique. PCA is an *unsupervised machine learning* technique that can be used to separate symmetric and asymmetric components, and has been adopted in our "ContextPCA" model, in order to reduce the context dimensions of the original data set. The experimental results on smartphone apps usage datasets show that "ContextPCA" model effectively predicts context-aware smartphone apps in terms of precision, recall, f-score and ROC values in various test cases.

**Keywords:** mobile data analytics; machine learning; principal component analysis; classification; decision tree; context-aware computing; user behavior modeling; predictive analytics; personalization; intelligent services; artificial intelligence; IoT applications

## 1. Introduction

Context-awareness is a popular term in the context of computing, because of the popularity of Internet of Things (IoT), particularly the recent advanced features in the most popular IoT device, i.e., smartphones. In the real world, users' interest on "Mobile Phones" is more and more than other platforms such as "Desktop Computer", "Laptop Computer" or "Tablet Computer" over time [1]. In addition to voice communication, people use smartphones for using various categories of apps like social networking system, tourist guide, shopping recommendation, instant messaging, medical appointment etc [2]. Users' behaviour with these apps may vary from user to user according to their contextual information in different dimensions such as temporal context, work status in workday or holiday, spatial context, their emotional state, Wifi status, or device related status etc. Although all these relevant contexts might have influence in apps usage behaviour of individuals, it may cause inefficient problem because of *higher dimensions* of contexts. Thus, it's important to study on *principal*

*component analysis* based on these contexts in order to build an effective and efficient context-aware apps prediction model.

Let us consider a real-world motivational example related to our ContextPCA model. Suppose, a smartphone user, Alice is a post graduate research student of Swinburne University of Technology. She has installed a large number of smartphone apps such as Facebook, LinkedIn, Twitter, Outlook email, Youtube, eHealth service, location tracking, instant messaging, read news etc. on her smartphone. Dynamic searching and efficiently finding these apps according to the needs in her various day-to-day situations would be useful. Although, homescreens of recent advanced smartphones provide easy access of the useful apps without additional searching effort, the homescreen is unaware about her current contexts, e.g., time. Consequently, the phone becomes unable to intelligently manage the useful apps according to her needs, as her current contexts may change over time. An efficient and effective context-aware apps prediction model could solve such problem and provide the required services. In the area of artificial intelligence and machine learning, tree-like model is one of the most popular approaches for predicting context-aware smartphone usage [3,4]. However, real-life phone usage data may contain *higher dimensions of contexts*, which may cause several issues like increases model *complexity*, may cause *over-fitting* problem, or decreases the model *prediction accuracy*. Thus, the research question is - *how to effectively minimize these issues while building a context-aware apps usage model?* Therefore, in this paper, we aim to focus on effectively reducing *higher dimensions of contexts* for building an intelligent context-aware smartphone *apps usage* predictive model based on machine learning techniques.

In the area of machine learning, there are typically two types of dimensionality reduction approaches such as feature elimination and feature extraction. In feature elimination approach, the features that are unnecessary are simply pruning from a dataset. We may lose any potential information gained from the dropped features. On the other hand, feature extraction creates new variables by combining the existing features and allows to maintain all important information held within features. As each contextual information might have an influence on individuals apps usage behaviour, we consider *feature extraction* approach rather than elimination. Principal components analysis (PCA) is an unsupervised, non-parametric statistical technique primarily used for dimensionality reduction in machine learning, that uses an orthogonal transformation which converts a set of correlated variables to a set of uncorrelated variables, which is briefly discussed in Section 3. It thus enables to identify correlations and patterns in a data set to transform into significantly lower dimension datasets without loss of any important information.

In this paper, we present an effective *principal component analysis (PCA)* based context-aware smartphone apps prediction model, *"ContextPCA"* using decision tree machine learning technique. In our earlier paper, we built an apps usage prediction model based on contexts [5]. Thus the key difference is focusing on *handling higher dimensions of contexts based on principal component analysis* in an apps usage prediction model. In our ContextPCA model, we first preprocess the raw apps usage datasets of individual users, that includes missing data handling, data encoding, and data scaling for further analysis. After that, we extract the contextual features from the training dataset based on principal component analysis and create a number of principal components that are less than the number of original context dimensions. Once the contexts have been processed into the principal components, we then construct a decision tree on the processed training dataset to achieve our goal. The effectiveness of producing different number of principal components and the ContextPCA model is studied through a number of experiments.

The contributions of this work can be summarized as follows.

- We first highlight the significance of *Principal Component Analysis (PCA)* for higher dimensions of contexts in a machine learning based context-aware smartphone apps usage prediction model.
- We have collected contextual *apps usage datasets* consisting of different categories of apps usages in different contexts that include both the *user-centric* context and *device-centric* context form individual smartphone users. We then analyse our collected apps usage datasets in terms of

context dimensions, in order to build a PCA-based context-aware prediction model *"ContextPCA"* using the decision tree classification approach.

- Finally, we conduct *experiments* to evaluate the effectiveness of different principal components in our ContextPCA model. The experimental results show that our ContextPCA model significantly outperforms for predicting context-aware smartphone apps.

The rest of the paper is organized as follows. Section 2 provides background and related work of machine learning classification approaches, and corresponding context-aware mobile services. Section 3 gives an overview of the principal component analysis. In Section 4, we present our ContextPCA model based on machine learning techniques. We have shown the experimental results on phone apps usage dataset in Section 5. Finally, Section 6 concludes this paper and highlights the future work.

## 2. Background and Related Work

To solve the prediction problems, classification learning is well-known and popular technique in the area of machine learning and data science. The goal of classification typically is to accurately classify or predict the given class labels of instances, whose contextual features or attribute values are known, but class values are unknown [6].

Although, association learning is another popular approach in the area of machine learning and data science and can be used for user behavioural analytics [7–11], we particularly focus on classification approach for the purpose of building a prediction model in this work. Classification learning techniques typically build the model using a given training dataset and then the resultant model can be used to predict the class label for a test case. Effectively modelling and predicting smartphone usage behaviour various machine learning techniques can be used. For instance, to build the prediction model in the area of mobile environment, ZeroR as base classifier, probability based naive Bayes classifier, support vector machines, instance based k-nearest neighbours, logistic regression, artificial neural network or deep learning, rule-based learning like decision trees, ensemble learning like random forest have been used [6,12]. These machine learning classifiers are frequently used in context-aware mobile analytics [12].

Among the traditional machine learning techniques, tree based context-aware model is more effective for predicting users' behavioural activities in different contexts [12]. A very well-known and mostly discussed tree based machine learning technique for building prediction model is decision trees [13]. ID3 algorithm proposed by Quinlan et al. is known as the core algorithm for building decision trees [14]. ID3 mainly constructs a top-down decision tree that follows a greedy searching procedure through the given training dataset. The entropy and information gain values are determined to select the best attribute or feature available in the datasets [14]. A modified algorithm named C4.5 algorithm proposed by Quinlan later, which is based on the ID3 algorithm [13]. This algorithm also builds decision trees using the concept of information gain mentioned earlier from a training dataset. Another gain based behavioral decision tree algorithm BehavDT has been proposed by Sarker et al. [4]. Decision tree classification approach is frequently used in the area of context-aware systems and services [3,15–17]. Recently, Sarker et al. use random forest ensemble learning technique consisting of multiple decision trees for predicting context-aware smartphone usage [5]. However, in that work, the authors particularly focus on how a single decision tree is affected by higher dimensions of contexts to build a context-aware model.

Unlike the above approaches and context-aware models, in this work, we present an effective *principal component analysis (PCA)* based context-aware smartphone apps prediction model, *"ContextPCA"* using decision tree machine learning technique. In this model, we aim to focus on reducing *higher dimensions of contexts* for building an effective context-aware smartphone *apps usage* predictive model based on machine learning techniques.

## 3. Principal Component Analysis

In the area of machine learning and data science, Principal Component Analysis (PCA) is a well-known unsupervised learning method. PCA is a mathematical procedure that transforms a number of correlated variables into a number of uncorrelated variables called principal components. PCA was at first presented in the area of non-arbitrary factors by Pearson [18] and reached out to irregular one by Hotelling [19]. Given a dataset having the number of dimensions $n$, PCA intends to find a linear subspace of dimension $d$, where $d < n$ such that the data points exist mainly on this linear subspace. Figure 1 shows an example of - how PCA effects on the dimensions of a given dataset. For instance, the original data instances have three features that are shown in Figure 1a with 3D space. After applying PCA, these data points can be reduced to two features shown in top of the Figure 1b, by projecting them onto a 2D plane with the principal components PC1 and PC2. Using PCA, the data can be further reduced to only one feature shown in bottom of Figure 1b, by projecting them onto a 1D line with the principal component PC1. The principal components mentioned above are orthogonal and linear transformations of the original data points, so that it could reduce the original dimensions $d < n$, in which we are interested in this PCA based context-aware smartphone apps usage model named "ContextPCA".


(**a**) Original features in 3D space. (**b**) Principal components in 2D and 1D space.

**Figure 1.** An example of a principal component analysis and corresponding components in different dimension space.

In the following, we summarize the basics of PCA including relevant mathematical equations. Lets consider $x_i$, $i \in 1...t$ a set of data vectors, PCA creates the $d$ principal axes based on those orthonormal axes onto which the variance retained under projection is maximal. This is known as the most common definition of PCA, due to Hotelling et al. [19]. In order to capture the variability as much as possible, we first choose $U_1$ as a principal component having maximum variance. Let the first principal component be a linear combination of $X$ defined by coefficients or weights $w = [w_1...w_n]$, and can be written in matrix form as $\mathrm{U}_1 = \mathrm{w}^T\mathrm{X}$. Thus the equation can be found as:

$$var\left(\mathrm{U}_1\right) = var\left(\mathrm{w}^T\mathrm{X}\right) = \mathrm{w}^T\mathrm{Sw} \tag{1}$$

where $S$ is the $n \times n$ sample covariance matrix of $X$ defined above. According to this equation $var\left(\mathrm{U}_1\right)$ can be made arbitrarily large by increasing the magnitude of $w$. Hence, above optimization problem is defined as maximization problem with respect to a constraint such that $max\ \mathrm{w}^T\mathrm{Sw}$ with respect to $w\mathrm{w}^T = 1$. To solve this optimization problem a Lagrange multiplier $\alpha_1$ is introduced and corresponding Lagrange function is constructed as:

$$L\left(w, \alpha\right) = \mathrm{w}^T\mathrm{Sw} - \alpha_1\left(\mathrm{w}^T\mathrm{w} - 1\right) \tag{2}$$

The solution of Equation (2) can be obtained by considering partial differentiation with respect to $w$ and $\alpha$ and further processing. Thus, the equations can be obtained as:

$$Sw = \alpha_1 w \tag{3}$$

$$\mathrm{w}^T \mathrm{S}\mathrm{w} = \alpha_1 \mathrm{w}^T \mathrm{w} = \alpha_1 \tag{4}$$

If $\alpha_1$ is the largest eigenvalue of $S$, then $var\,(\mathrm{U}_1)$ is maximized. Based on the equations involved $\alpha_1$ and $w$ are an eigenvalue and an eigenvector of $S$. Differentiating Equation (2) with respect to the Lagrange multiplier $\alpha_1$ results constraint as:

$$\mathrm{w}^T \mathrm{w} = 1 \tag{5}$$

Thus, it has been shown that the normalized eigenvector with the largest associated eigenvalue of the sample covariance matrix $S$ gives the first principal component. A similar argument can show that the first $d$ principal components are determined by the $d$ dominant eigenvectors of covariance matrix $S$.

## 4. Materials and Methods

### *4.1. Contextual Data Collection and Description*

In this work, we take into account a number of contexts including not only the user-centric context such as spatio-temporal context of the users, their mood or preferences, etc., but also the device-centric context considering users' influences for their usage. Hence, we summarize these contextual information that are used in our ContextPCA model. These are:

*Smartphone apps:* In this work, different categories of smartphones' apps such as social networking, instant messaging, mobile communications, entertainment, or other apps related to users' daily life services are considered in order to build our ContextPCA model. For instance, Figure 2 shows the distributions of different types of apps like Facebook (FB), Gmail, LinkedIn (LI), Instagram (IG), Youtube (YT), Live Sport (LS), Whatsapp (WA), Browsing (BR), Movie (MOV), Skype (SK), Music (MS), Read News (RN), Games (GM) of a sample user.

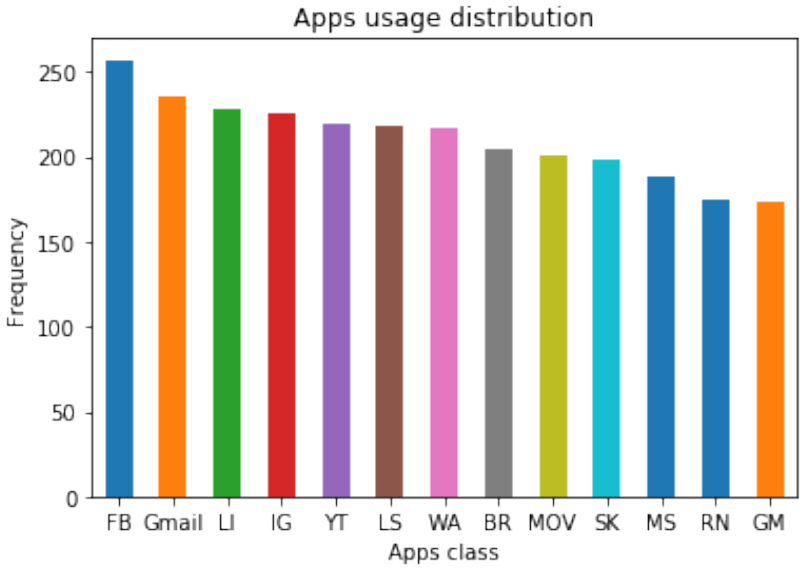



**Figure 2.** Smartphone apps usage distribution of a sample user considering various types of apps.

*Contexts:* According to the general definition of context, it could be anything to characterize the situation of an entity. In this work, a smartphone user can be represented as an entity. Thus, different dimensions of information might have an influence on the apps usage of smartphone users. For instance, *temporal context* that represents time related information of the users' apps usage behaviour. Temporal context is one of the significant and primary contexts that have highly influence on smartphone users for their activities with their phones [20]. In addition to such temporal information,

users' *work status* could be another context heavily impacts on apps usage for many individuals. For instance, apps usage behaviour of an individual user on Saturday, say a holiday, may differ with her usage on Monday, a first working day in a week. Although, it is related to the temporal context in terms of week day and week end, however, it also represents individuals' working status, which is significant context in order to model smartphone apps usage behaviour according to their preferences. *Spatial context* could be another significant context that represents users' spatial information, e.g., one's current location at office. As spatio-temporal context is popular for building human-centric context-aware applications, such spatial context could play a role in our smartphone apps usage behaviour model. *User mood* could be another significant context that impacts on individuals, particularly on human centric applications. For instance, one individual user typically likes to listen only her top favourite musics when her emotional state is in a happy mood, while likes to chatting with her close friends on social media when her emotional state is in a sad mood.

Besides these user-centric contexts that are related to users' day-to-day situations or personal preferences, users' own device related contexts also important for modelling users' apps usage behaviourr. Such contextual information could be one's phone profile, phone battery level or charging status etc. that might have an influence on users to use various categories of smartphone apps. For instance, if one's device gives low power signal, she typically might not be interested to connect her device with the Internet in that context for using an entertainment app like watching Youtube video. For modelling users' apps usage behaviour *Internet connectivity* and speed might also have an impact in our real world life. Thus, in this work, we consider all these contextual information in our PCA based modelling. We have summarized the detailed picture of the contexts that are used in our ContextPCA model in Table 1. To collect these contextual information from individual users, we have randomly chosen ten participants and collected their datasets from June 2018 to October 2018 for the purpose of doing experiments.

**Table 1.** An overview of contexts in our ContextPCA model.

| Contexts | Type | Example Values |
|---|---|---|
| Temporal Context | Continuous | Time-of-the-day [24-h-a-day]<br>Days-of-the-week [7-days-a-week] |
| Spatial Context | Categorical | Phone user location [at home, at office,<br>at the canteen, in the playground,<br>on the way, etc.] |
| Work status Context | Categorical (binary) | Workday and Holiday |
| User mood Context | Categorical | Emotional state of phone user [normal, happy, or sad] |
| Device status Context | Categorical | Battery level[low, medium, or full] |
| Phone profile Context | Categorical | Phone notification [general, silent, or vibration] |
| Internet connectivity Context | Categorical (binary) | WiFi connectivity [on, off] |
| Smartphone apps | Categorical | Social networking, Gmail, Communication, Video,<br>Entertainment, Read News, Games etc. |

### *4.2. Preprocessing of Contextual Data*

To build our ContextPCA model, we need exploratory data analysis collected by us to feed our target machine learning classification technique. In this procedure below tasks are involved for this work.

- *Missing data handling:* In our datasets, we found only a few number of missing data that occurs during the data collection process. Thus, due to anomaly raised in contextual data collection, we first remove all the missing data and consider the relevant contextual features and corresponding data-values.
- *Contextual feature encoding:* As we have seen that Table 1 contains contextual information including categorical context. In order to fit these data to the machine learning based model, it is needed to convert all the categorical contextual features into vectors. To do this task, the most common

approaches are "Label Encoding" and "One Hot Encoding". In one hot encoding technique, a significant number of features increases, and consequently increases the data dimensions. On the other hand, in label encoding, the number of features remains the same as the feature-values directly converted into a specific numeric values. As we have taken into account a variety of contexts discussed above, one hot encoded features might have sparse data which could makes the model inefficient in terms of processing time because of handing additional high dimensions of data. Thus, in this ContextPCA model, we consider label encoding technique rather than one hot encoding in our pre-processing task. Lets consider an example in terms of context user mood. Label encoding can turn user diverse mode [happy, sad, normal, happy, sad] into vectors [0, 1, 2, 0, 1] representing numeric values.

- *Feature scaling:* In data processing, it is also known as data normalization. Feature scaling is a method used to normalize the range of independent variables or contextual features of data. We use Standard Scaler that normalizes the features with the mean = 0 and standard deviation = 1.

### *4.3. PCA-Based Decision Tree Generation*

Once the preprocessing of contexts has been completed, we generate a PCA based decision tree in order to build ContextPCA model. To build a PCA based decision tree, we use the principal components rather than using all the contexts discussed above. For this, we first create a number of principal components based on principal component analysis discussed above. It thus enables to identify correlations and patterns in a data set to transform into significantly lower dimension datasets without loss of any important information. After generating the principal components, we employ the most popular machine learning algorithm decision tree on the generated components [13]. Decision tree algorithm builds decision tree from a training dataset, using the concept of entropy and information gain [13].

In terms of structure, a decision tree builds a tree-like model that includes a root node from where the tree starts top-down growing, a number of internal or interior nodes that represent the test cases, and corresponding leaf nodes that are generated for representing the outcome of these tests. Each interior node in our ContextPCA model denotes a context-aware test case on a particular condition, and each leaf node represents the corresponding outcome of that test which is represented by a category of apps or class label (e.g., using Facebook app). Each branch in the tree are connected with arcs, from root node to leaf node. These leaf nodes are also known as terminal nodes as the tree stops to grow after finding a leaf node. Once the tree has been built, it is used to predict each test instance. For this, it generates a number of IF-THEN logical rules and classify them. Overall, there are two basic steps for the development of our decision tree based ContextPCA model; (a) building the decision tree from a apps usage training dataset considering multi-dimensional contexts, and (b) applying the generated decision tree to measure the prediction accuracy of the context-aware test cases.

## 5. Experimental Results and Discussion

In this section, we first highlight the evaluation metrics that are taken into account to evaluate our ContextPCA model, and discuss the experimental results in various dimensions related to our analysis.

### *5.1. Evaluation Metric*

To evaluate our ContextPCA model, we employ the most popular K-fold cross validation technique in machine learning [6]. In our evaluation, we use $K = 10$ for generating train and test data to build model and measure the predicted accuracy in terms of precision, recall, and F-score. If TP, FP, FN denote true positives, false positives, and false negatives respectively, then the formal definition of these metrics are as below [21]:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$Fscore = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{8}$$

In addition to these metrics, we also take into account ROC (Receiver Operating Characteristic) that summarizes the trade-off between true positive rate and false positive rate for a machine learning based predictive model [21].

### *5.2. Explained Variance of Principal Components in our ContextPCA Model*

In this experiment, we show the explained variance of the principal components and the effect of this variance in our ContextPCA model. The fraction of variance explained by a principal component is the ratio between the variance of that principal component and the total variance. For this, Figure 3 shows the explained variance for various components utilizing the datasets of two different users. Principal component analysis in our ContextPCA model computes a new set of variables ("principal components") and expresses the data in terms of these new variables, such as $PC_1$, $PC_2$, $PC_3$, $PC_4$, $PC_5$, $PC_6$, and $PC_7$ shown in Figure 3. The new variables known as principal components generated are able to represent the similar information as the original variables, and can be considered as the transformed one by taking into the relevant variances.

If we observe Figure 3, the highest fraction of explained variance among these variables is 23% for User-01 and 22% for User-02. The lowest one is 6% for User-01 and 7% for User-02. We can also compute these fractions for the subsets of principal components. For instance, $PC_1$ and $PC_2$ together explains 39% for User-01 of the total variance, and 38% for User-02. Similarly, first three components $PC_1$, $PC_2$, and $PC_3$ together explain 54% for User-01 of the total variance, and 54% for User-02, and so on. Figure 4 shows the cumulative graph considering all the principal components $PC_1$, $PC_2$, $PC_3$, $PC_4$, $PC_5$, $PC_6$, and $PC_7$ and their explained variances for User-01 and User-02 respectively utilizing their apps usage datasets.

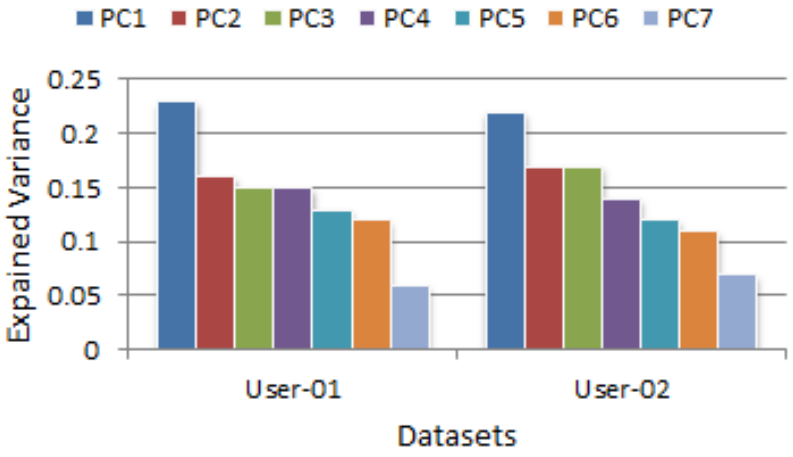
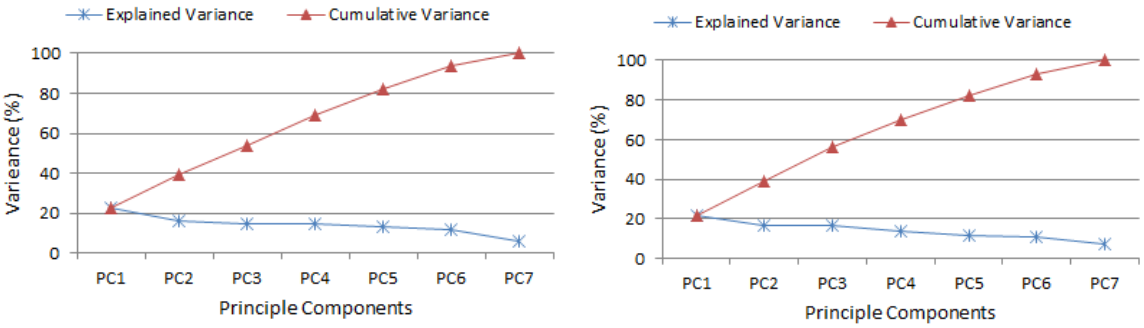



**Figure 3.** Explained variance for different principal components generated in our ContextPCA model utilizing the apps usage datasets of User-01 and User-02 respectively.




(**a**) Variance for User-01. (**b**) Variance for User-02.

**Figure 4.** Cumulative variance for different users.

### *5.3. Prediction Results of our ContextPCA Model*

In this experiment, we show the prediction results of our ContextPCA model. For this, Tables 2 and 3 show the prediction results in terms of Precision, Recall, and F-score. In this experiment, we have shown the results for each combination of the principal components $PC_1$, $PC_2$, $PC_3$, $PC_4$, $PC_5$, $PC_6$, and $PC_7$. If we observe Tables 2 and 3, we see that the first component $PC_1$ contains more significant information to predict apps usage behaviour. By adding additional components like $PC_2$ increases the prediction results. However, more components might not be significant in terms of further processing speed and prediction results as well. According to the results shown in Table 2, a cumulative variance generated for the combination $PC_1$, $PC_2$, $PC_3$, $PC_4$, $PC_5$ is optimal to get the better prediction results for User-01. An optimal is determined with higher prediction results with lower components. Similarly, a cumulative variance generated for the combination $PC_1$, $PC_2$, $PC_3$, $PC_4$, $PC_5$, $PC_6$ is optimal to get the better prediction results for User-02.

**Table 2.** Prediction results for various combinations of components for User-01.

| Components | Precision | Recall | F-Score |
|---|---|---|---|
| PC1 | 0.85 | 0.85 | 0.85 |
| PC1, PC2 | 0.86 | 0.86 | 0.86 |
| PC1, PC2, PC3 | 0.87 | 0.87 | 0.87 |
| PC1, PC2, PC3, PC4 | 0.88 | 0.88 | 0.88 |
| PC1, PC2, PC3, PC4, PC5 | 0.89 | 0.89 | 0.89 |
| PC1, PC2, PC3, PC4, PC5, PC6 | 0.89 | 0.89 | 0.89 |
| PC1, PC2, PC3, PC4, PC5, PC6, PC7 | 0.89 | 0.89 | 0.89 |

**Table 3.** Prediction results for various combinations of components for User-02.

| Components | Precision | Recall | F-Score |
|---|---|---|---|
| PC1 | 0.86 | 0.85 | 0.85 |
| PC1, PC2 | 0.87 | 0.87 | 0.87 |
| PC1, PC2, PC3 | 0.87 | 0.87 | 0.87 |
| PC1, PC2, PC3, PC4 | 0.88 | 0.87 | 0.87 |
| PC1, PC2, PC3, PC4, PC5 | 0.88 | 0.87 | 0.87 |
| PC1, PC2, PC3, PC4, PC5, PC6 | 0.89 | 0.88 | 0.88 |
| PC1, PC2, PC3, PC4, PC5, PC6, PC7 | 0.87 | 0.87 | 0.87 |

In addition to these components based overall results, we have also shown individual class wise prediction results for a particular combination of principal components for User-01, shown in Table 4. The results are shown using the optimal subsets of the principal components for the first five components $PC_1$, $PC_2$, $PC_3$, $PC_4$, $PC_5$ for User-01. If we observe Table 4, we see the significant results for each class as well. Thus, from the overall experimental results shown in Tables 2 and 3,

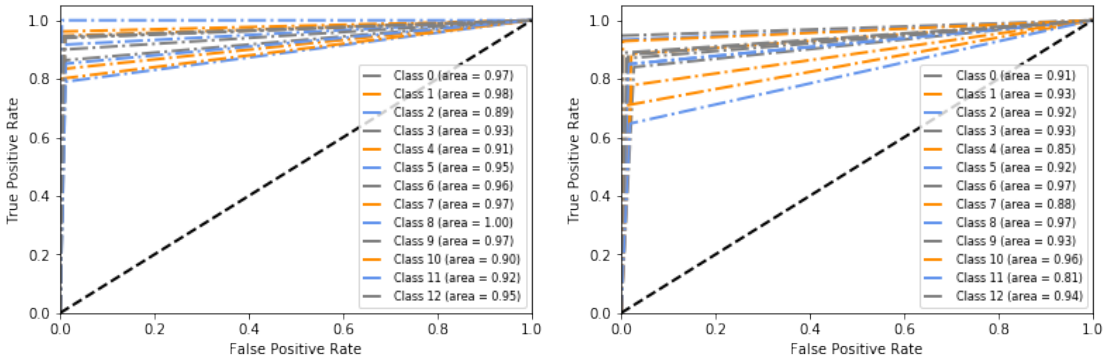
and individual class wise experimental results shown in Table 4, we can conclude that our ContextPCA model is able to effectively predict each app usage behaviour class of individual users according to their usage patterns in the datasets.

**Table 4.** Individual class wise prediction results for a particular combination of principal components for User-01.

| Class | Precision | Recall | F-Score |
|---|---|---|---|
| Class 0 | 0.90 | 0.93 | 0.92 |
| Class 1 | 0.85 | 0.92 | 0.88 |
| Class 2 | 0.92 | 0.80 | 0.85 |
| Class 3 | 0.93 | 0.93 | 0.93 |
| Class 4 | 0.89 | 0.89 | 0.89 |
| Class 5 | 0.95 | 0.93 | 0.93 |
| Class 6 | 0.88 | 0.89 | 0.89 |
| Class 7 | 0.88 | 0.91 | 0.90 |
| Class 8 | 0.89 | 0.89 | 0.89 |
| Class 9 | 0.88 | 0.95 | 0.91 |
| Class 10 | 0.86 | 0.84 | 0.85 |
| Class 11 | 0.89 | 0.85 | 0.87 |
| Class 12 | 0.86 | 0.86 | 0.86 |

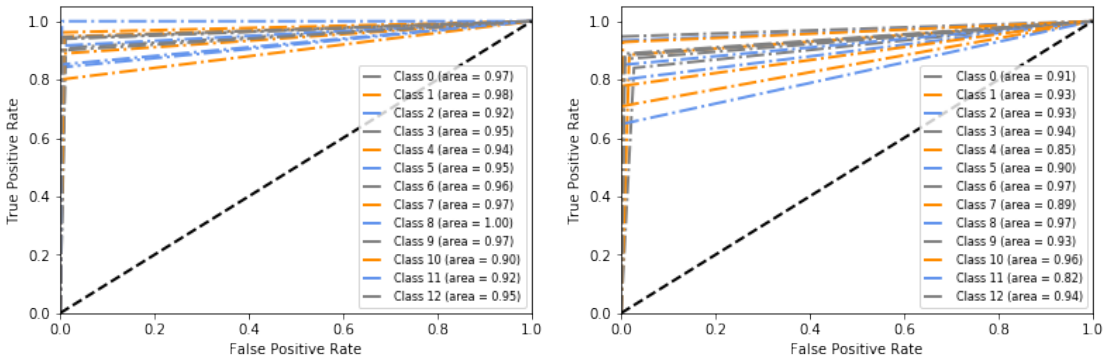
### *5.4. ROC Analysis of our ContextPCA Model*

In this experiment, we compute and compare the effectiveness in terms of ROC of our context-aware model ContextPCA for each individual class. To show the effectiveness for individual users, Figure 5 shows the ROC values for two different individuals User-01 and User-02 utilizing their own datasets. In addition to this results, we have also shown the ROC values considering multiple decision trees in our model. In this experiment, we consider 10 decision trees rather than single decision tree while building the model. For this we randomly divide the datasets into 10 sets and build a single decision tree utilizing each set of data and finally merge the results. Figure 6 shows the ROC values for each individual class considering such multiple decision trees in our ContextPCA model. Thus, the overall experimental results shown in Figures 5 and 6, we can conclude that our ContextPCA model is able to effectively predict each app usage behaviour class of individual users according to their usage patterns in the datasets.




(**a**) ROC utilizing User-01 dataset. (**b**) ROC utilizing User-02 dataset.

**Figure 5.** Prediction results of our ContextPCA model in terms of ROC for each individual class.




(**a**) ROC utilizing User-01 dataset. (**b**) ROC utilizing User-02 dataset.

**Figure 6.** Prediction results of our ContextPCA model considering multiple decision trees in terms of ROC for each individual class.

### *5.5. Effectiveness Comparison and Discussion*

In this experiment, we compute and compare the effectiveness of our context-aware model ContextPCA, with the traditional decision tree model. As our model is personalized, we show the comparing results for individual users selected randomly. Figure 7 shows the relative comparison of prediction results in terms of precision, recall, f-score for two different individuals User-01 and User-02 utilizing their apps usage datasets. In addition to these individual results, we also show the average results utilizing a collection of datasets of all ten users, in Figure 8.

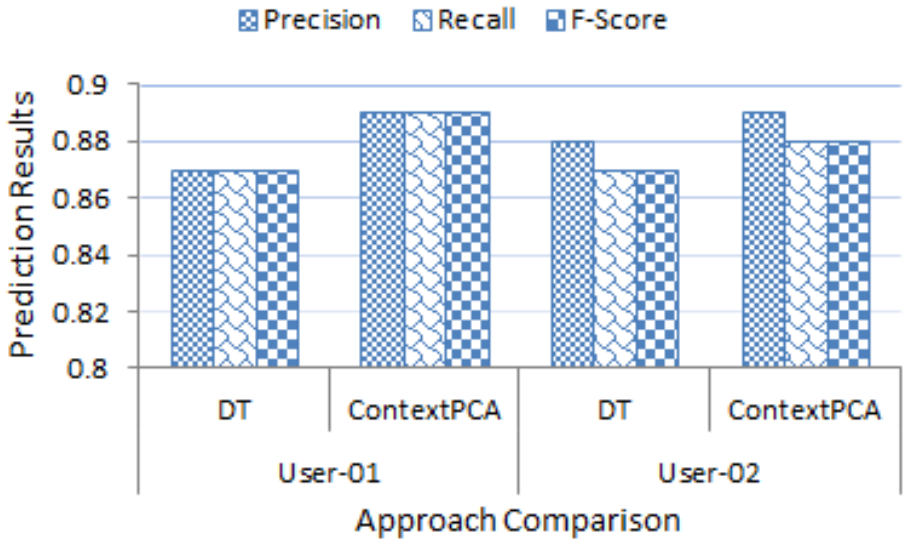



**Figure 7.** Comparing prediction results with a traditional tree-based machine learning model in terms of precision, recall, and f-score utilizing individual user's dataset.

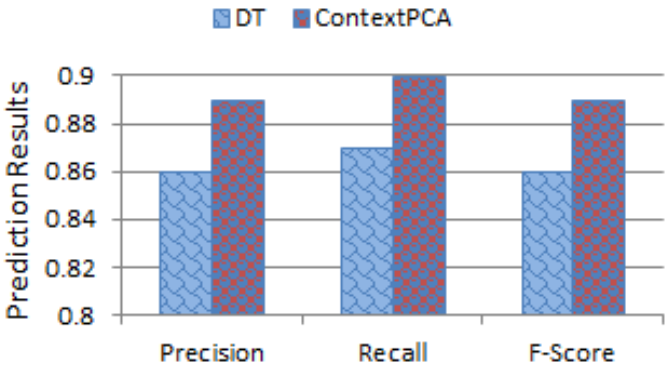



**Figure 8.** Comparing prediction results with a traditional tree-based machine learning model in terms of average precision, recall, and f-score utilizing a collection of datasets.

If we observe Figures 7 and 8, we see that our ContextPCA model gives better prediction results in terms of precision, recall, and f-score than traditional decision tree model for each individual user's dataset. The reason is that the decision tree model creates a decision tree by considering all the contexts available in the datasets. As we claim that considering higher dimensions of contexts may cause over-fitting problem and consequently decrease the prediction accuracy. Moreover, considering higher dimensions of contexts while building the decision tree model increases the complexity. On the other hand, we take into account principal component analysis for handling the higher dimensions of contexts. Thus, our ContextPCA model minimizes the overfitting problem and reduces the complexity while designing the tree like model and improves the accuracy as well.

In addition to the above results, Figure 9 shows the predicted outcome comparing with classic machine learning algorithms. For this comparison purpose, we select several basic and popular machine learning algorithms that are frequently used in the area of mobile analytics. These are ZeroR, naive Bayes classifier (NBC), support vector machine (SVM), and k-nearest neighbor (KNN). According to Figure 9, ContextPCA model outperforms these traditional machine learning algorithms as well when applying on contextual mobile phone data. Thus, based on the discussion on experimental results, we can conclude that the ContextPCA model could be more effective, when huge datasets with high dimensions of contexts are available.

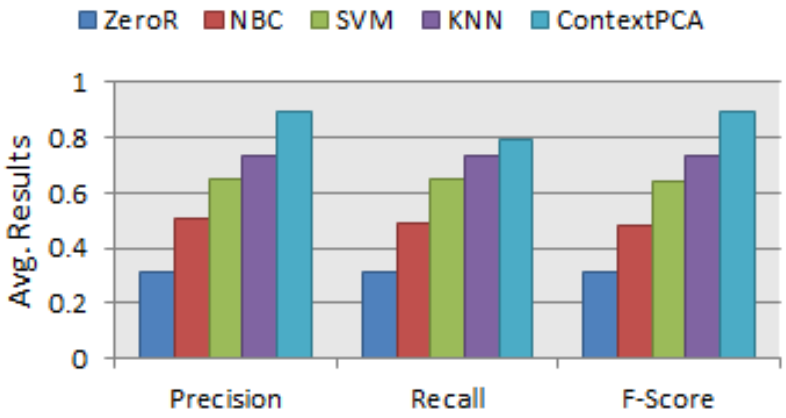



**Figure 9.** Comparing prediction results with traditional machine learning techniques in terms of average precision, recall, and f-score utilizing a collection of datasets.

## 6. Conclusions and Future Work

In this paper, we have presented an effective principal component analysis based context-aware smartphone apps prediction model, ContextPCA using decision tree machine learning technique.

In our ContextPCA model, we have adopted PCA to reduce the context dimensions of the original data set by producing a new set of uncorrelated components, to make the model effective and efficient. In order to build this data-driven ContextPCA model, we have taken into account a number of contextual features that might have an influence on users' apps usage in their various day-to-day real world situations, and collected corresponding apps usage datasets from smartphone users. No assumption or prior knowledge is needed in employing our ContextPCA model as we take into account unsupervised learning technique PCA for feature extraction and supervised decision tree for building the model based on the principal components generated. Experimental results on the datasets indicate that our ContextPCA model outperforms while predicting individuals' smartphone apps. We believe that this ContextPCA model would be helpful to application developers to build corresponding real-life applications for the end users, particularly, where higher dimensions of contexts involved.

To assess the effectiveness of our ContextPCA model by collecting more dimensions of contextual data in the domain of smart cities and Internet-of-Things, and to measure the effectiveness in application level could be a future work.

**Author Contributions:** All authors contributed and have read and agreed to publish this manuscript.

**Funding:** This research was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under Grant No. D-166-611-1441.

**Acknowledgments:** This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under Grant No. D-166-611-1441. The authors, therefore, gratefully acknowledge DSR technical and financial support.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Sarker, I.H. Context-aware rule learning from smartphone data: survey, challenges and future directions. *J. Big Data* **2019**, *6*, 1–25.
2. Sarker, I.H. Mobile Data Science: Towards Understanding Data-Driven Intelligent Mobile Applications. *EAI Endorsed Trans. Scalable Inf. Syst.* **2018**, *5*, e4. [CrossRef]
3. Sarker, I.H. A machine learning based robust prediction model for real-life mobile phone data. *Internet Things* **2019**, *5*, 180–193. [CrossRef]
4. Sarker, I.H.; Colman, A.; Han, J.; Khan, A.I.; Abushark, Y.B.; Salah, K. BehavDT: A Behavioral Decision Tree Learning to Build User-Centric Context-Aware Predictive Model. *Mob. Netw. Appl.* **2019**, 1–11. [CrossRef]
5. Sarker, I.H.; Salah, K. AppsPred: Predicting Context-Aware Smartphone Apps using Random Forest Learning. *Internet Things* **2019**, *8*, 1–11. [CrossRef]
6. Han, J.; Pei, J.; Kamber, M. *Data Mining: Concepts and Techniques*; Elsevier: Amsterdam, The Netherlands, 2011.
7. Zhu, H.; Chen, E.; Xiong, H.; Yu, K.; Cao, H.; Tian, J. Mining Mobile User Preferences for Personalized Context-Aware Recommendation. *ACM Trans. Intell. Syst. Technol.* **2014**, *5*, 58. [CrossRef]
8. Sarker, I.H.; Salim, F.D. Mining User Behavioral Rules from Smartphone Data through Association Analysis. In Proceedings of the 22nd Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Melbourne, Australia, 3–6 June 2018; pp. 450–461.
9. Sarker, I.H.; Colman, A.; Han, J. RecencyMiner: Mining recency-based personalized behavior from contextual smartphone data. *J. Big Data* **2019**, *6*, 1–21. [CrossRef]
10. Sarker, I.H. Research issues in mining user behavioral rules for context-aware intelligent mobile applications. *Iran J. Comput. Sci.* **2018**, *2*, 41–51. [CrossRef]
11. Agrawal, R.; Srikant, R. Fast algorithms for mining association rules. In Proceedings of the International Joint Conference on Very Large Data Bases, Santiago, Chile, 12–15 September 1994; Volume 1215, pp. 487–499.
12. Sarker, I.H.; Kayes, A.; Watters, P. Effectiveness Analysis of Machine Learning Classification Models for Predicting Personalized Context-Aware Smartphone Usage. *J. Big Data* **2019**, *6*, 1–28. [CrossRef]
13. Quinlan, J.R. C4.5: Programs for Machine Learning. In *Machine Learning*; Morgan Kaufmann Publishers, Inc.: Burlington, MA, USA, 1993.

14. Quinlan, J.R. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
15. Zulkernain, S.; Madiraju, P.; Ahamed, S.I.; Stamm, K. A Mobile Intelligent Interruption Management System. *J. UCS* **2010**, *16*, 2060–2080.
16. Hong, J.; Suh, E.H.; Kim, J.; Kim, S. Context-aware system for proactive personalized service based on context history. *Expert Syst. Appl.* **2009**, *36*, 7448–7457. [CrossRef]
17. Lee, W.P. Deploying personalized mobile services in an agent-based environment. *Expert Syst. Appl.* **2007**, *32*, 1194–1207. [CrossRef]
18. Pearson, K. On lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **1901**, *2*, 559–572. [CrossRef]
19. Hotelling, H. Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.* **1933**, *24*, 417. [CrossRef]
20. Sarker, I.H.; Colman, A.; Kabir, M.A.; Han, J. Individualized Time-Series Segmentation for Mining Mobile Phone User Behavior. *Comput. J. Oxf. Univ.* **2018**, *61*, 349–368. [CrossRef]
21. Witten, I.H.; Frank, E.; Trigg, L.E.; Hall, M.A.; Holmes, G.; Cunningham, S.J. *Weka: Practical Machine Learning Tools and Techniques with Java Implementations*; University of Waikato: Hamilton, New Zealand, 1999.