

Recognizing Falls and Surfaces Using Mobile Devices

Vicente J.P. Amorim², Carlos A. L. Mion¹, Igor M. Pereira¹,
Ricardo C. Camara², Andrea G. C. Bianchi², Ricardo A. R. Oliveira²

¹iMobilis Lab – Department of Computer and Systems (DECSI)
Federal University of Ouro Preto (UFOP)
João Monlevade – MG – Brazil

²Department of Computing (DECOM)
Federal University of Ouro Preto (UFOP)
Ouro Preto – MG – Brazil

Abstract. *Mobile devices are getting much more relevance during the users' day, in a way that they are paying to increase device security and durability through external cases or insurance plans. However, these approaches are useless if the individuals does not properly take care of their devices. This paper describes an approach to monitor and classifies a surface where a smartphone falls, making possible to categorize this crash into a range of dangerousness. The authors collected empirical data from device falls to make possible the development of an optimal classifier. Our results reached up to 88% recognition rate of surfaces considering a specific features subset, letting us conclude that it is possible to infer user care level through the analysis of how a device is being treated.*

1. Introduction

Currently, the number of active mobile devices worldwide has already surpassed the amount of people [Independent 2014]. In some countries, there are already more than three devices per person [Portal 2014]. Projections expect this number still increasing about five times population growing rate annually. By the users' point of view, many personal mobile devices require more caution and care when handling them. With large screens surrounded by plastic cases, these pieces of equipment commonly need extra protection given by accessories, such as reinforced external cases, glass films for the screens, and so on. Even though, unsatisfied users appeal to other solutions to keep safe their devices, such as protection and insurance plans.

Despite many options to keep devices protected, users' care still the most important point to maintain the hardware safe. Both insurance plans and physical accessories do not provide a focus on user behavior, mainly if devices carelessness is a recurring conduct. No protection plan or the external case will guarantee device sanity without users proper care. Indeed, especially in the enterprise context, care with lend equipment is critical, with companies providing devices to their employees and expecting that they will return it safe and functional. Moreover, this same uncertainty scenario can be found when considering parents and their sons. Then, it is easy to note that a deeper monitoring mechanism, which takes into account the final users' behavior, will help devices owners to keep their hardware safe. Besides that, insurance companies may have a higher guarantee that users are not abusing their benefits.

Some related proposals try to recognize user behavior considering mobile devices inertial sensors, such as accelerometer and gyroscope. Data coming from these sensors may allow special applications infer whether a device/user is falling to the ground or not [Majumder et al. 2013][Mehner et al. 2013]. Although imperative, knowing when a device/user is falling is not enough by itself without having surface dangerousness estimation. Another set of solutions tries to infer the surface characteristics where a device is placed [Darbar and Samanta 2015][Cho et al. 2012].

This work presents a novel classification method that allows inferring the dangerousness of a surface where a mobile device falls. Through the use of device accelerometer sensors, the authors collected data from falls and classified the surfaces into four different categories. Results presented at the end allow us to conclude that it is possible to detect a reckless user behavior. This information can be managed for user self-monitoring or even to inform devices real owners. Briefly, the key contributions of this paper are: an Android platform application that continuously monitors accelerometer data triggering for a fall; and a classification method to distinguish between four different surfaces. This classification may allow other mobile applications aiming to characterize users care level when handling their mobile devices.

This paper is divided into the following segments: Section 2 present the works related to the one described here. Section 3 describes fundamental concepts that should be taken into account to understand this paper. Section 4 depicts main actions taken during the development phase. Section 5 outline the results reached by this work, while Section 6 presents the general conclusion and provides an overview of future works.

2. Related Works

Researches using data gathered from mobile device's sensors can be found in the literature. Especially inertial sensors, such as accelerometers and gyroscopes, have been contextualized with a lot of different solutions. The majority of related works focus on the analysis of users and devices falls besides surface type recognition. Although no identical work was found in literature, this section will present bellow proposals that are, in some way, related to the one described here.

A fall detection algorithm is depicted by [Mehner et al. 2013]. This solution is based on a threshold that uses acceleration data to trigger for a fall. The work presents a fall detection description, including the stages before and after a fall occurrence. However, this work does not provide any information about falls severity regarding the surface where object hits. Similarly, the work presented in [Majumder et al. 2013] describes the "iPrevention" system. Built inside a smartphone, it is in charge to detect elderly patients falls identifying high-risk ones. Despite the presented results, this system also does not deal with surface analysis as a way to characterize the risk of a fall.

Another class of related works focuses on surface recognition using data from mobile devices sensors. Proposals described in [Darbar and Samanta 2015] and [Hwang and Wohn 2013] focus on the physical users' context, presenting different systems to identify locations where their mobile devices are placed, such as pant's pocket, backpack and wooden table. On both works, there is no analysis connecting the device context and surface with falls detection. Moreover, [Cho et al. 2012] shows the "VibePhone" solution, that focuses on the same concepts as the previous works. An

SVM (Support Vector Machine) is used to classify vibration data obtained from an accelerometer that varies according to the surface where a smartphone is placed. By its turn, [Windau and Shen 2010] describes a system for surface identification based on the time spent until inertia. A robot arm “knocks” the surface and analyzes accelerometer data to characterize its parameters, such as hardness, elasticity and stiffness. In [Sinapov et al. 2011], a similar approach is considered using a robot hand to rub different surfaces, analyzing generated vibrations through the use of SVM and k-nearest neighbors (k-NN) classifiers. However, these two previous proposals are not feasible to be applied on a mobile device once they require external equipment.

Finally, work presented by [Weiss et al. 2006] uses a vehicle context to describe the surface recognition problem. A car prototype (cart) containing sensors was used to collect data after runs on seven different scenarios. The proposed idea is similar to the one considered in this paper. However, collected data is different once a fall is not considered.

3. Background

To properly understand the work described in this paper, some basic concepts must be introduced. Three key concepts used here are presented bellow.

3.1. Accelerometer Sensor

Accelerometer sensor is in charge of measuring acceleration values applied over it through three main vectors (X,Y,Z), that describe the acceleration magnitude related to each direction. Any linear combination of these three vectors is also possible to be obtained. This sensor was used in this work to gather the acceleration values over time during a fall, and after that, when the mobile device hits the ground. Considering this, the accelerometer sensor can be used to measure gravity acceleration based on the values of three axes. Figure 1 depicts an accelerometer vectors inside a generic smartphone.

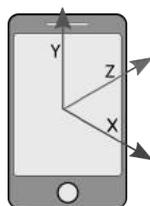


Figure 1. Accelerometer vectors applied to a generic device.

3.2. Surfaces Dangerousness

The surface roughness can imply a higher or a lower risk to the mobile device integrity, not only the screen, but also inner hardware. In this work, the roughness classification was the focus in a way that it can be used in the calculation of how much damage the device may suffer.

3.3. Classification Methods

Bellow is presented a short description about the variety of classification methods used in this work:

- **Logit Boost:** Initially proposed by [Friedman et al. 1998]. This class performs classification using a regression scheme as the base learner. Can handle multi-class problems;
- **Bagging:** Initially proposed by [Breiman 1996], this method does bagging to a classifier to reduce variance. Can do classification and regression depending on the base learner;
- **Random Forest:** Initially proposed by [Breiman 2001], the Random Forest method (or “Random Decision Forests”) operate by constructing a multitude of decision trees at training time. It outputs a class that is used as a kind of mean prediction (regression) of the individual trees. Random decision forests correct for decision trees’ habit of overfitting to their training set; and
- **Support Vector Machine (SVM):** Initially proposed by [Boser et al. 1992], the Support Vector Machine (SVM) is a supervised learning classification algorithm that focuses on a training set to separate new elements into existing groups. In its commonest version, SVM classifies the items into two different sets. However, there are algorithm versions considering a multi-label approach, classifying data into several groups. It is valid to note that development phase takes into account an existent SVM implementation: LibSVM [Chang and Lin 2011], which supports multiset/multilabel approach (second block of “Data Pre-Processing” phase on Figure 2).

4. Development Considerations

The development of solution presented by this paper was separated into four steps: Environment preparation, data acquisition, data pre-processing and training process, and classification. Except for environment preparation step, Figure 2 presents an overview about every development stage, detailed on the remaining of this section.

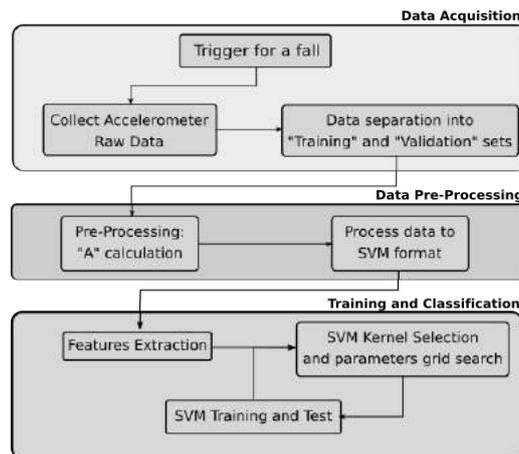


Figure 2. Steps taken by development process to classify surfaces of a fall.

4.1. Environment Preparation

Several tests were conducted aiming to develop and validate an application that identifies a surface where a mobile device falls. Four different surfaces were considered: sand, grass, towel extended on the ground and a pillow. These surfaces were taken into account

because they represent the high and low-end points regarding dangerousness for a falling device: Here, to allow tests repetition without permanent damage to the smartphone, a towel extended on the ground was considered as the most critical surface, while a pillow was the less dangerous one.

4.2. Data Acquisition

This work has used an Android test application, which was developed by the authors, installed on smartphone devices to gather accelerometer sensor data. This application works sampling data at a normal rate (controlled by Android “SENSOR_DELAY_NORMAL” constant) until a fall is detected. Once it happens, the system starts to collect information in a higher frequency (controlled by Android “SENSOR_DELAY_FASTEST” constant). The whole process can be described according to the following steps:

1. **Trigger for a fall:** Constantly analyzes a time window of samples and searches for an acceleration value that indicates a device fall;
2. **Collect accelerometer raw data:** Record data from accelerometer sensor right after it hits the ground and while the sensor do not stabilize; and
3. **Data separation:** Separate collected data into two different sets (training and validation). Data from one set are not used by the other one.

For every group, 400 runs were done (100 falls for each surface), in a total of 800 runs. Moreover, data acquisition was made taken into account the following parameters:

- Height of fall: 85cm from the highest point to the top of the surface. Calculated as the average height of a common pant’s pocket;
- Mobile devices positioned on top and at the edge of a platform with previously described height; and
- Two different mobile devices – smartphones – used: Samsung Galaxy Note 2 and LG E977.

4.3. Data Pre-Processing

As previously described in Subsection 3.1, accelerometer sensor gets the values associated with three different vectors (X, Y and Z). These vectors are in charge to measure acceleration imposed at each of the axes using m/s^2 unit. Still, data collected by the previous step was obtained in the raw format, i.e., no pre-processing was applied. Still, to ease the training and classification process, these three axes acceleration were converted to a single value (A) that describes full acceleration applied to the device (first block of “Data Pre-Processing” phase on Figure 2). “A” value calculation, which is the absolute value of X, Y and Z variables, from accelerometer axes, can be given through the following equation: $A = \sqrt{X^2 + Y^2 + Z^2}$.

Figure 3 shows the behavior of each acceleration vector during a smartphone fall. It is possible to observe that “X, Y and Z” values were encoded inside the “A” line. Additionally, it is easy to note three generic phases during a fall: free fall; bouncing and stabilization. In fact, these three parts are used here by the classifiers as key points to characterize the surface where the devices hit. In the end, “A” value becomes constant and remains around gravity acceleration after smartphone stabilization.

Moreover, after conversion of “X, Y and Z” into “A”, data was organized in the format accepted by each classifier.

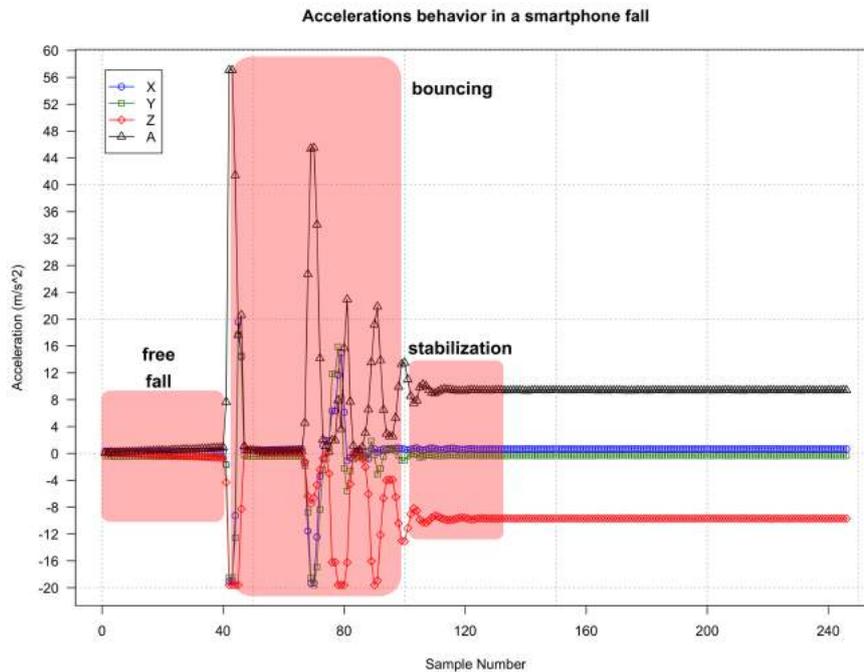


Figure 3. Behavior of each acceleration vector during a fall.

4.4. Training Process and Classification

Last development step is related to training and classification processes. To properly classify the input data, it must be organized concerning inner knowledge it can indirectly provide. Pre-processed information cannot be directly used by classifiers once it has a significant amount of data, requiring more time to train and classify. Instead, features should be extracted from the dataset to reduce their size and increase their representativeness.

As depicted by Figure 3, which presents the behavior of a fall in a pillow surface, it is possible to find three different phases: free fall, bouncing and stabilization. These three steps help to identify a type of surface where a device hits (the more the device bounces, the more smooth is the surface). This behavior allowed the authors to raise a set of 57 different features. Every one of these features was inferred considering the whole dataset and also specific information. The majority of them were calculated from simple statistical dispersion measures: mean, median, standard deviation, and so on, applied to the data obtained from the device fall, such as: the number of bounces, the highest bounce amplitude and the time until stabilization. Such characteristics played a major role helping classifiers to differentiate from a surface to another one. Most relevant features gathered from the dataset were:

- **Time Until Stabilization:** As the number of bumps varies according to the surface where device hits, time until stabilization will also change from surface-to-surface. Then, it was calculated and used as a feature for data classification;
- **Data Dispersion of Samples Until Stabilization:** A set of statistical dispersion values were calculated (mean, standard variation, variance, median, min value and max value) and used as features. Use of these values helps to describe the behavior of free fall and bounce phases data;

- **Data Dispersion of Fast Fourier Transform (FFT) and Signal Magnitude:** FFT was used to analyze the signal frequency domain. After that, the FFT spectrum magnitude was calculated and its statistical dispersion values used as another feature set. Figure 4 presents a graph describing a frequency domain of one of device falls;
- **Smoothness Rate:** Although it was already considered, this feature also treats maximum signal amplitude. The smoothness rate index is given by this value divided by the time (in number of samples) until device stabilization; and
- **Amplitude Histogram:** Here the amplitude spectrum was divided into several slots. As presented by Figure 5, each slot represents the number of samples found at each amplitude. It is valid to note that, the majority of samples are concentrated around the value of $9.8m/s^2$. This happens because most time the “A” value is susceptible only to the gravity acceleration.

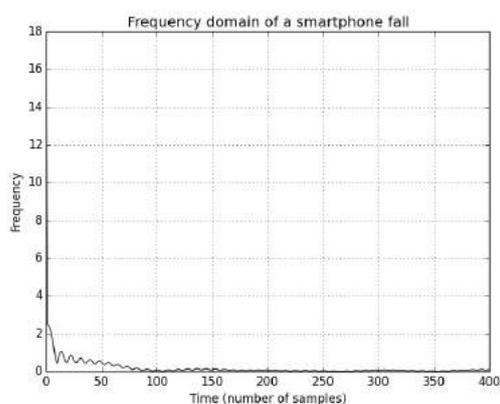


Figure 4. Smartphone fall frequency domain analysis using Fast Fourier Transform (FFT).

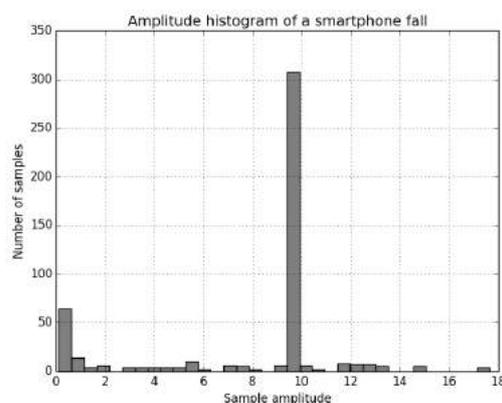


Figure 5. Smartphone fall histogram amplitude analysis.

In the end, every fall data was parsed to extract previously listed features. Nevertheless, the same approach was applied to “Training” and “Validation” sets of samples.

After “Features Extraction” step, training and classification phases proceeded to classifier configuration and input parameters grid search. Both steps are associated, once input parameters choice depends on which classifier configuration was chosen. For every configuration a grid search has been executed, initially with a wide search space and then, with a specific search trying to refine parameters input values.

This phase final step was the application of selected constraints on each classifier training and validation sets. Here, this cycle was also refined through the repetition of input parameters selection, grid search execution, training and validation until the best result was obtained.

5. Empirical Results

The previous section presented the steps taken by this work, from raw data collection to surface classification. This section will outline main results found by this research, considering that 200 data instances were collected (100 for training group and 100 for

validation group) from four different surfaces: sand, grass, towel extended on the ground and pillow.

One of the main challenges faced by this work was smartphone behavior during a fall. In related works, such as [Darbar and Samanta 2015][Cho et al. 2012] and [Weiss et al. 2006], the accelerometer sensor was statically positioned on top of considered surfaces. Here, a not so controlled environment was found. Accelerometer sensor inside the device has been continuously moving according to the fall trajectory. This behavior is particularly difficult to classify, once it may vary, or not, according to the position that smartphone touches the surface when falling. For instance, if the smartphone touches the ground with one of its edges, the trajectory thereafter may be totally different from a fall when the device touches the surface with its screen.

In addition to the features extracted from data (subsection 4.4), the classifiers were also trained and validated using raw data, i.e., with no type of features extracted. However, the results obtained using this approach were not as good as that one using extracted features. All the results presented here consider the multi-label classification approach, where an instance of a device fall was classified evaluating it against four possible surfaces. Moreover, extracted features have been used with the same weight. Bellow, it is described the main results as well as a comparison between different training approaches and classifiers configurations.

5.1. Logit Boost

Best result using Logit Boost classifier was found considering the following input parameters: **Likelihood threshold:** -1.798; and **Number of iterations:** 10. Table 1 presents the number of positive and negative recognition rate considering only “validation” set of instances.

Table 1. Classification results using Logit Boost.

	Sand		Grass		Towel		Pillow	
	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.
Positive	70,0%	30,0%	82,0%	18,0%	66,0%	34,0%	68,0%	32,0%
Negative	9,7%	90,3%	17,0%	83,0%	7,7%	92,3%	3,7%	96,3%
Accuracy	0,853		0,828		0,858		0,893	
Precision	0,707		0,617		0,742		0,861	
Recall	0,700		0,820		0,660		0,680	
F-score	0,704		0,704		0,698		0,759	

5.2. Bagging

Best result using Bagging classifier was found considering the following input parameters: **Number of decimal places:** 2; and **Number of iterations:** 10. Table 2 presents the number of positive and negative recognition rate considering only “validation” set of instances.

5.3. Random Forest

Best result using “Random Forest” classifier was found considering the following input parameters: **Number of execution slots:** 1; and **Number of iterations:** 100. Table 3

Table 2. Classification results using Bagging.

	Sand		Grass		Towel		Pillow	
	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.
Positive	73,0%	27,0%	80,0%	20,0%	62,0%	38,0%	73,0%	27,0%
Negative	11,7%	88,3%	12,3%	87,7%	8,0%	92,0%	5,3%	94,7%
Accuracy	0,845		0,858		0,845		0,893	
Precision	0,676		0,684		0,721		0,820	
Recall	0,730		0,800		0,620		0,730	
F-score	0,702		0,737		0,667		0,772	

presents the number of positive and negative recognition rate considering only “validation” set of instances.

Table 3. Classification results using Random Forest.

	Sand		Grass		Towel		Pillow	
	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.
Positive	69,0%	31,0%	88,0%	12,0%	72,0%	28,0%	60,0%	40,0%
Negative	7,3%	92,7%	18,0%	82,0%	7,6%	92,4%	4,0%	96,0%
Accuracy	0,868		0,835		0,873		0,870	
Precision	0,758		0,619		0,758		0,834	
Recall	0,690		0,880		0,720		0,600	
F-score	0,723		0,727		0,738		0,698	

5.4. SVM with Radial Basis Function Kernel (RBF)

Best result using SVM RBF kernel was found considering the following input parameters: **SVM type (s)**: C-SVC; **Cost (c)**: 52.0; and **Gamma (γ)**: 0.00000765. Table 4 describes the number of positive and negative recognition rate considering only “validation” set of instances.

Table 4. Classification results using SVM RBF kernel.

	Sand		Grass		Towel		Pillow	
	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.
Positive	76,0%	24,0%	79,0%	21,0%	62,0%	38,0%	59,0%	41,0%
Negative	13,7%	86,3%	15,7%	84,3%	4,3%	95,7%	7,7%	92,3%
Accuracy	0,837		0,830		0,872		0,840	
Precision	0,649		0,627		0,827		0,719	
Recall	0,760		0,790		0,620		0,590	
F-score	0,700		0,699		0,708		0,648	

5.5. SVM with Linear Kernel

Best result using SVM linear kernel was found considering the following input parameters: **SVM type (s)**: C-SVC; and **Cost (c)**: 0.055. Table 5 presents the number of positive and negative recognition rate considering only “validation” set of instances.

Table 5. Classification results using SVM linear kernel.

	Sand		Grass		Towel		Pillow	
	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.
Positive	79,0%	21,0%	86,0%	14,0%	72,0%	28,0%	79,0%	21,0%
Negative	8,7%	91,3%	10,0%	90,0%	6,3%	93,7%	3,0%	97,0%
Accuracy	0,882		0,890		0,882		0,925	
Precision	0,752		0,741		0,791		0,898	
Recall	0,790		0,860		0,720		0,790	
F-score	0,771		0,796		0,754		0,840	

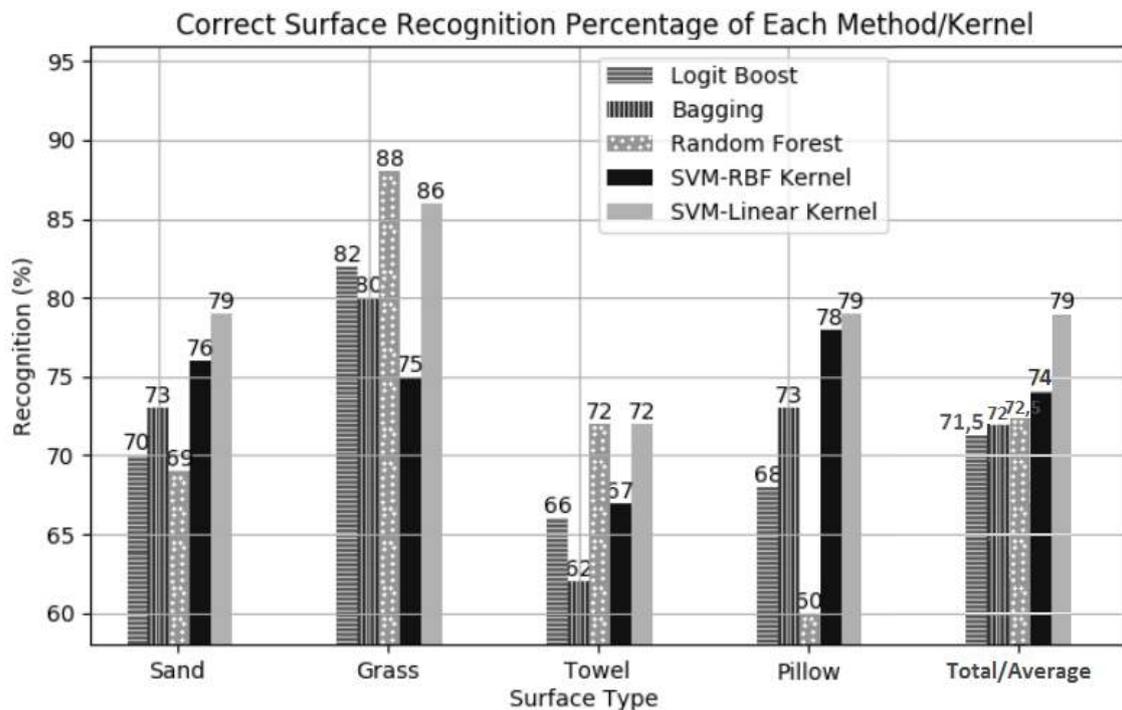
**Figure 6. Comparison between all the classifiers separated by surface types.**

Figure 6 presents a comparison between surface recognition rate of each class. It is easy to note that SVM with linear kernel had better results on average. Still, it also had an excellent performance on grass surface when compared to most results. Moreover, a general behavior can be seen when analyzing each surface separately. On grass, all the classifiers had an acceptable classification rate (true positives), a fact that might indicate that grass surface data might be easily differentiated from other surfaces. Conversely, the “towel extended on the ground” reached the lowest true positive rates, indicating that this surface is usually misclassified. The same behavior can be observed even on classifiers that had good rates on other surfaces, pointing out that the “towel on the ground” signal was classified as belonging to another surface.

When analyzing performance measures, it is possible to verify that results presented in Tables 1, 2, 3, 4 and 5 are endorsing measured true positive and false negative ratios. Given values of precision, accuracy and recall are high even when comparing SVM linear kernel with other classifiers. Furthermore, values found for F-score guarantees a

good precision/recall relation, indicating an acceptable data reliability level.

6. Conclusion and Future Works

Users are increasingly seeking for a way to protect their hardware. External cases and insurance plans are considered as ways to boost device protection. Despite their efficiency, these approaches do not provide a method to quantify the user care level with their hardware. For instance, a company may want to know if a lend smartphone is being handled with care by its employees.

Furthermore, several researches using mobile devices have been conducted today. Most of these studies are focused on energy efficiency or mobility aspects, with a segment emphasizing the users and their context, taking device sensors information as input. Inside this last class of researches, it is possible to find works that try to infer the surfaces where devices are placed, such as “VibePhone” [Cho et al. 2012] and “SurfaceSense” [Darbar and Samanta 2015].

The work presented here outlines an approach for classifying a smartphone fall as a way to monitor users care/careless level with their devices. As described by results, in some scenarios, we reached up to 88% recognition rate considering four different surfaces. Despite this value could initially be seen as low, it is necessary to take into account the whole environment where the user is inserted. Here, the main objective was to identify and categorize a repetitive user behavior that over time could result in a device damage or even a crash. Finally, in the case of this attitude becomes repetitive, a notification might be sent to the hardware owner, warning about user conduct.

As a way to increase system robustness, future works may enclose tests and validation of a broad range of devices, including new brands and screen sizes. Additionally, more surfaces can also be taken into account, trying to estimate user behavior dangerousness better.

Acknowledgment

The authors would like to thanks the iMobilis Laboratory, PROPP/UFOP and CNPq for the financial support and provided infrastructure.

References

- Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92*, pages 144–152, New York, NY, USA. ACM.
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.
- Chang, C.-C. and Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27.
- Cho, J., Hwang, I., and Oh, S. (2012). Vibration-based surface recognition for smartphones. In *Proceedings of the 2012 IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, RTCSA '12*, pages 459–464, Washington, DC, USA. IEEE Computer Society.

- Darbar, R. and Samanta, D. (2015). Surfacesense: Smartphone can recognize where it is kept. In *Proceedings of the 7th International Conference on HCI, IndiaHCI 2015, IndiaHCI'15*, pages 39–46, New York, NY, USA. ACM.
- Friedman, J., Hastie, T., and Tibshirani, R. (1998). Additive logistic regression: a statistical view of boosting. Technical report, Stanford University.
- Hwang, S. and Wohn, K. (2013). Vibrotactor: Low-cost placement-aware technique using vibration echoes on mobile devices. In *Proceedings of the Companion Publication of the 2013 International Conference on Intelligent User Interfaces Companion, IUI '13 Companion*, pages 73–74, New York, NY, USA. ACM.
- Independent (2016). There are officially more mobile devices than people in the world. <http://www.independent.co.uk/life-style/gadgets-and-tech/news/there-are-officially-more-mobile-devices-than-people-in-the-world-9780518.html>. [Online: accessed 16-June-2016].
- Majumder, A. J. A., Rahman, F., Zerín, I., Ebel, Jr., W., and Ahamed, S. I. (2013). iprevention: Towards a novel real-time smartphone-based fall prevention system. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13*, pages 513–518, New York, NY, USA. ACM.
- Mehner, S., Klauck, R., and Koenig, H. (2013). Location-independent fall detection with smartphone. In *Proceedings of the 6th International Conference on Pervasive Technologies Related to Assistive Environments, PETRA '13*, pages 11:1–11:8, New York, NY, USA. ACM.
- Portal, S. T. S. (2014). Average number of connected devices used per person in selected countries in 2014. <http://www.statista.com/statistics/333861/connected-devices-per-person-in-selected-countries/>. [Online: accessed 16-June-2016].
- Sinapov, J., Sukhoy, V., Sahai, R., and Stoytchev, A. (2011). Vibrotactile recognition and categorization of surfaces by a humanoid robot. *IEEE Transactions on Robotics*, 27(3):488–497.
- Weiss, C., Frohlich, H., and Zell, A. (2006). Vibration-based terrain classification using support vector machines. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4429–4434.
- Windau, J. and Shen, W. M. (2010). An inertia-based surface identification system. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2330–2335.