ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 12 (2022) No. 2, pp. 497-511
DOI: 10.33168/JSMS.2022.0227

Mobile App Usage Behaviour based User Authentication using Fuzzy Random Forest

Kavyashree Nagarajaiah[1], Supriya Maganahalli Chandramouli[2], Lokesh Malavalli Ramakrishna[3]

[1]Dr.Ambedkar Institute of Technology, Bengaluru, Karnataka, India
[2]Sri Siddhartha Institute of technology, Tumkur, Karnataka, India
[3]Maharaja Institute of Technology Mysore, Mandya, Karnataka, India

kavyashree1283@gmail.com

Abstract. Smartphones are common among people and are also used for monitoring and controlling house appliances. Identity verification on smartphones is typically done once and the identification method is not repeatedly used. Extraction of data from mobile telephone usage, including authentication and authorization events can be used to find a resolution. In this method, Fuzzy logic with the Random Forest user authentication model is presented for a mobile application that uses events of application access. The real-world dataset has been employed to test the approach. The framework is tested for authenticated users using shared applications for a particular time. The proposed method considers that the users operate the mobile apps individually during weekend days. The proposed method is tested with alternative classifiers in terms of the identity of valid users. The proposed model has a higher F-measure of 99.98 % compared to the existing K-Nearest Neighbor has 96 % F-measure.

Keywords: Fuzzy logic, F-measure, mobile, random forest, user authentication

1. Introduction

Many end devices, including mobile phones and tablets, provide internet access and also include identification methods, like fingerprint scanners, for gaining access to a device or associated apps (Xu et al., 019). After initial login, many users avoid configuring brief access periods. As a result, devices following the entry point are vulnerable to unauthorized use. As a result, it's critical to propose a solution that addresses these concerns and may be used implicitly and continually in the backdrop (Garcia-Martin et al., 2019). Cell devices use access control behavior for authorization events and authentication to arrive at solutions (Stanik et al., 2019). The benefit of user behavior patterns identification is that a lot of information may be gathered and used for continuous identity verification, such as apps, device capabilities, and infrastructure components created while using apps on mobile devices (Amalfitano et al., 2019). In addition, the number of applications used on mobile smartphones is growing that occupies more resources. For example, there are currently over two million smartphone applications accessible in the major application store, and more are being released regularly. In this research, many behavior patterns authentication systems that use wearable smart and phone applications have been released (Shahidinejad et al., 2020).

Thabtah et al., (2020)offer an energy model creation technique that analyses power usage per application on devices using built-in battery charge sensors. Nevertheless, when other programmers are operating in the background, it is difficult to predict power usage exclusively for individual apps. The researchers demonstrated that by analyzing texts, calls, and emails, abnormalities can be found due to its interaction with mobile apps (Junior et al.,). This research proposes a user profile strategy that collects behavioral data like contacts, texts, and geographical regions. The existing method presents an unusual occurrence detection strategy monitoring and measuring users' behaviors, such as sending a text message or making phone calls, using machine learning. An authenticating strategy based on application usage behavior is presented in another study (Arslan et al., 2019). The App description, activity name, text length, and also emails, call length, and date, were all used in this investigation. Application usage statistics are used to develop a continuous authentication method. The study doesn’t examine all individuals' use of applications and considers specific users (Mattson et al., 2020). This research proposes a behavior monitoring approach that uses previous app usage to continually verify phone devices. The period span is limited by 22 days. The strong authentication methods based on app access permissions and produced traffic while using these applications are described in (Islamiati et al., 2019). The existing methods of software behaviour analysis doesn’t consider account shared apps, weekends access times, and different weekdays. In this paper a classifier system based on an ensemble of fuzzy decision trees, i.e., a fuzzy random forest, is proposed which combines the robustness of multiple classifier systems, the power of the randomness to increase the diversity of the trees, and the flexibility of fuzzy logic and fuzzy sets for imperfect data management.

In this paper, the literature review is presented in Section 2 and the proposed methodology is explained in Section 3. The experimental results and findings are explained in Section 4 and the conclusion of this paper is presented in Section 5.

2. Literature Review

The existing methods based on software mobile applications are reviewed in this Section. The advantages and limitations of the reviewed method are also described in this Section.

Yosef Ashibani (2019) designed an Authentication and authorization events were used to create a machine learning strong authentication framework for mobile technology systems. During weekend days, the model was validated for ongoing authenticated users who are using common apps at the same daily duration. In addition, alternative classifiers are evaluated in terms of valid user identity. It's vital to note that the number of individuals was investigated was quite minimal. As a result of the limited sample size, every individual may exhibit a wide range of behavior.

Melih Iphar (2019) created a mobile app based on multi decision-making methods for deep mining techniques. Created software platform method to complete selection process based on a reasonable alternative set prioritizing using mining technique. The developed method considers various choice elements and doesn’t consider the traditional underground coal technique. Mobile application creation relevance narrowing m-learning demand and high possession amount of mobile gadgets.

Sabiha Yeni (2020) created a mobile app is created based on multi decision-making methods for deep mining techniques. Created software platform method to complete selection process based on a reasonable alternative set prioritizing using mining technique. The developed method considers various choice elements and doesn’t consider the traditional underground coal technique. Mobile application creation relevance narrowing m-learning demand and high possession amount of mobile gadgets.

R. Srivatsan (2020) employed a machine learning-based predictive model and smartphone app software platform for forecasting COVID-19 disease vulnerability using health information were described. Individuals may use the proposed machine learning technique to take extra steps to protect themselves against COVID-19 transmission, and physicians and government authorities can target the most vulnerable individuals to keep the epidemic at bay. The accuracy suggests that there could be certain examples of people who do not have greater sensitivity but are yet classified as high-risk.

Liangyi Gong (2020) attempted Implementing Machines Training on a Large-Scale Malicious Mobile Detection System. It identifies Android malware by analyzing their run-time usage of a limited number of carefully chosen APIs, which is supplemented by other information such as eligible and utilized intentions. The developed method utilized bit vectors to represent features extracted in the current architecture, which would be compact and fast in practice but may miss important features extracted (e.g., API invocation frequencies) and result in an over-fitting problem.

3. Methodology

This presents an authentication approach based on machine learning, which learns users' app accessing event patterns and authenticates them based on the authentication pattern. Fig. 1 illustrates the architecture of the proposed Fuzzy logic with the Random Forest model for mobile software applications.

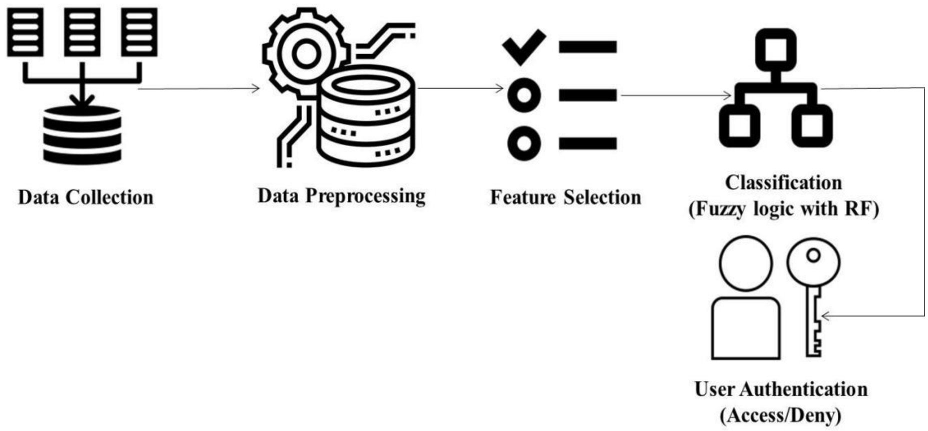

Fig. 1: The system architecture of proposed fuzzy logic with random forest method for mobile application

3.1. Data collection

During this stage, the system creates an access history of the users of the app who are using it and selects the appropriate features for them. A smart home hub captures information about app usage history as a result of training and authentication and also it captures the user id, date, and package name.

3.2. Data preprocessing

Data is first collected and then preprocessed for training; app access logs are received by the model and they are preprocessed in addition to taking into account any imbalanced class representation among the observations. This study considers the day of the week, duration, access time, user ids, app names, and class variance, based on previously accessed apps as inputs and takes its data in the numerical series manner. The dates where the data is missing are filled through average value of corresponding attribute of data. Assume unequal variances to measure student t-test that is used to measure the significant difference between weekend apps, weekday apps, and two sample sets.

3.3. Classification

To select the most reliable classifiers, fuzzy logic with a random forest classifier is utilized after preprocessing and selecting the features. The training data is set as 70 and the testing data is set as 30.

3.3.1. Fuzzy logic system

Despite the fact that decision tree algorithms have proven to be interpretable, efficient, and capable of coping with big datasets, they are notoriously unstable when small perturbations are introduced into training datasets. Fuzzy logic has been introduced into decision tree construction techniques as a result of this. Fuzzy logic provides a remedy to alleviate instability by leveraging its inherent elasticity. Existing techniques have successfully coupled fuzzy sets and their underlying approximation reasoning capabilities with decision trees. The fuzzy set integration of random forest equips the classifier with the benefit of uncertainity management with comprehensibility of linguistic variables, and popularity and easy application of decision trees.

The resulting trees have improved noise resistance, a broader application to uncertain or ambiguous circumstances, and support for the tree structure's comprehensibility, which remains the primary representation of the generated information. As a result, a random forest has been proposed, with a fuzzy decision tree as the basic classifier. The researchers picked random forest over the other ensemble strategies based on decision trees because, like boosting, it produces the best outcomes. Moreover, the random forest has more resistant capability when the proportion of any attribute of data sample of a class in the training set is randomly altered than other boosting based ensemble methods.

Hence, the proposed method takes the advantage of improvement in results as multiple classifier system than the individual classifier and increase the noise resistance property of random forests based ensemble methods to incorporate fuzzy decision trees in them.

Let X = {X1, . . . , XF } be the set of input variables and XF +1 be the output variable. Since we consider classification problems, XF +1 is a discrete variable, which can assume values in {C1, . . . , CM}, where M is the number of possible classes. Let Uf , with f = 1, ..., F, be the universe of the f th variable Xf . Let TTR = {(x1, xF +1,1), . . . ,(xN , xF +1,N)} be the tree training set composed of N input-output pairs, with xp = [x1,p . . . , xF,p] and xF +1,p ∈ {C1, . . . , CM}. Input variables

Xf can be continuous or categorical. Continuous variables need to be partitioned for generating the decision tree.

Unlike the fuzzy decision tree learning used, we do not assume that continuous variables are partitioned before starting the tree learning, but we determine these fuzzy partitions during the tree generation. We aim to propose an approach that is easy to implement, is computationally light and guarantees to achieve accuracy values comparable with classical random forests. The ratio behind this approach is to explore specific zones of the input domain more and more in detail during the tree generation. In practice, we adopt a sort of zoom in on this specific zones. The "magnifying glass" is a strong fuzzy partition consisting of three triangular fuzzy sets. We adopt this partition because it is determined by just choosing a point, the core of the intermediate triangular fuzzy set.

More formally, for each attribute Xf, we sort the attribute values xf,1 . . . , xf,NS for the set S. Let lf and uf be the lower and upper bounds of the universe in Xf of the points contained in S. To determine the optimal cut-point tf for variable Xf, we pose tf in correspondence of the f-th coordinate (except lf and uf) of each point of the universe. For each possible candidate, we define a strong fuzzy partition of the universe by using three triangular fuzzy sets, namely Af,1, Af,2 and Af,3. The cores of Af,1, Af,2 and Af,3 coincide with lf, tf and uf, respectively.

In classical crisp decision trees, the points can belong to only one of the subsets generated by partitioning the example set of the parent node. In fuzzy decision trees, with strong partitions, one point xp can belong to two different fuzzy sets, for instance B1 and B2, with complementary membership degrees (μB1 (xi) = 1 − μB2 (xi)). To simplify the generation of the tree and to reduce its deepness, we consider in the child nodes only the examples with membership degree higher than 0.5. Thus, for each fuzzy set in the partition, we create a node and transfer to this node the examples which belong to the α − cut, with α = 0.5, of the fuzzy set. We verified that this choice simplifies the generation of the fuzzy decision tree without affecting the final accuracy.

Then, we compute the points af,1−2 and af,2−3, where the membership function (MF) of Af,1 intersects the MF of Af,2, and the MF of Af,2 intersects the MF of Af,3, respectively. More precisely,

$$af,1-2 = lf + tf\ 2 \tag{1}$$

$$af,2-3 = tf + uf\ 2 \tag{2}$$

Let S1, S2 and S3 be the subsets of examples in S with values of input variable Xf lower than or equal to af,1−2, larger than af,1−2 and lower than af,2−3, and larger than or equal to af,2−3, respectively. We recall that the cardinality of a fuzzy set is defined as

$$|S| = X\ NS\ i=1\ \mu S(xi) \tag{3}$$

where NS is the number of objects in S. From this definition, we compute the cardinality of Sj , j = 1, 2, 3 as:

|Sj | = XNj i=1 µSj (xi) = XNj i=1 TN(µAf,j (xf,i), µS(xi)) (4)

where Nj is the number of values of Xf (crisp cardinality) in the set Sj, µSj (xi) = TN(µAf,j (xf,i), µS(xi)) is the membership degree of example xi to set Sj , µAf,j (xf,i) is the membership degree of example xi to fuzzy set Af,j , µS(xi) is the membership degree of example xi to set S (for the root of the decision tree, µS(xi) = 1) and the operator TN is a T-norm. In the experiments, we adopted the product as T-norm.

3.3.2. Random forest

The Random Forest method is an ensemble learning algorithm that builds small, computationally feasible decision trees of a few features and then combines them to form a strong learning process. Each tree in the woods will vote on the class for the item based on the information vector value. Once all the trees have voted, the info vector will order another item. Every single tree in the forest is ranked according to the order with the most votes. The forest builds each tree as follows:

- Let the number of models in the first preparing information be N. Draw a bootstrap test of size N from the first training information. This example will be another training dataset for developing the tree. The first training information is not in the bootstrap test called out-of-pack information.
- Let the absolute number of info highlights in the first training information be M. On this bootstrap test information, just m ascribes are picked indiscriminately for each tree where m < M. The credits from this set makes the most ideal split at every hub of the tree. The worth of m ought to be steady during the development of the forest.

The exactness of the individual trees and the relationship between the trees in the forest decide the blunder pace of the forest. While expanding the connection builds the forest blunder rate, expanding the precision of the individual tree diminishes the error rate of the forest. The lessening m decreases both the relationship and the strength. Contrasted with a solitary choice tree calculation helps RF runs productively on huge datasets with a superior exactness. RF can deal with ostensible information and doesn't over-fit. For a given training dataset T, the Gini Index can be expressed as:

$$\sum\sum(f\ (C_i,T)(f\ (j,T) \qquad (1)$$

Where $f\ (C_i,T)$ the probability that a selected case belongs to class C_i. Thus, by using a given combination of features, a decision tree is made to grow up to its maximum depth. The ultimate choice for characterization of test information is finished by greater part casts a ballot from expectations of the outfit of trees. The pictorial representation of the RF model is shown in Fig. 2.

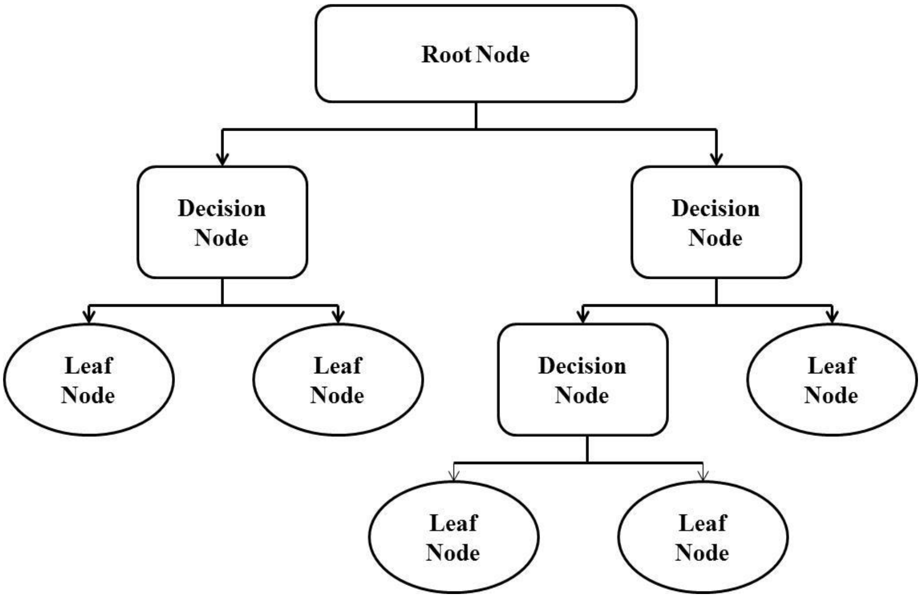

Fig. 2: The pictorial representation of the random forest model

With this learning method, each tree in the forest is analyzed separately for a given input vector. After each tree has a classification model, the method selects the class with the highest frequency. Individual trees of low error rate are present in the forest that decrease the overall error rate of the model. If two trees are correlated with the third tree, the error rate will also decrease. Furthermore, the RF method is fast compared to another classifier model, and trees with less correlation have a lower error rate. Each tree in the forest has a different bootstrap sample. A tree T construction doesn't consider some parts of original data and remaining data are used for test T. If the number of trees in the forest is not increased, then the overfitting problem is not developing in the model. The RF algorithm performs well based on the number of trees in the forest. One method is applied for weekdays and weekends for selected classification methods in the dataset. Many trees decrease the performance of classification accuracy and an iterative method is used to optimize these parameters.

In the random forest, each tree is constructed to the maximum size and without pruning. During the construction process of each tree, every time that it needs to split a node (i.e. select a test at the node), only a random subset of the total set of available attributes is considered and a new random selection is performed for each split. The size of this subset is the only significant design parameter in the random forest. As a result, some attributes (including the best) might not be considered for each split, but an attribute excluded in one split might be used by other splits in the same tree. Random forests have two stochastic elements: (1) bagging is used for the selection of the datasets used as input for each tree; and (2) the set of attributes considered as candidates for each node split. These randomizations increase the diversity of the trees and significantly improve their overall predictive accuracy when their outputs are combined. When a random forest is constructed, about 1=3 of the examples are excluded from the training dataset of each tree in the forest. These examples are called ‘‘out of bag” (OOB); each tree will have a different set of OOB examples. The OOB examples are not used to build the tree and constitute an independent test sample for the tree.

To compute the fuzzy information gain, we exploit the weighted fuzzy entropy. Let F Ent(Sj), j = 1, 2, 3, be the fuzzy entropy of Sj defined as:

F Ent(Sj) = XM m=1 − |Sj,Cm| |Sj | log2(|Sj,Cm| |Sj |) (5)

where Sj,Cm is the set of examples in Sj with class label equal to Cm. Then, the weighted fuzzy entropy W F Ent(tf ; S) is computed as:

W F Ent(tf ; S) = X 3 j=1 |Sj | |S| F Ent(Sj) (6)

The fuzzy information gain F Gain for variable Xf is defined as:

F Gain(A, tf ; S) = F Ent(S) − W F Ent(A, tf ; S). (7)

As in the discretization approaches used in crisp decision trees, a continuous variable can be considered in several decision nodes in the same path from the root to a leaf. In each node, we apply the same fuzzy partition to the universe of the set of objects that belong to the node with membership value higher than 0. For instance, let us assume that continuous variable Xf is used in a decision node to partition the universe [lf , uf]. The partition generates three child nodes, which contain objects with values of variable XF in h lf , lf +tf 2 i , h lf +tf 2 , tf +uf 2 i and h tf +uf 2 , uf i , respectively. Let us suppose that in the path generated from the first child node another decision node considers variable Xf . Then, a new partition is generated by considering the universe h lf , lf +tf 2 i . In practice, the new partition is devoted to analyze in detail a specific subset of the initial Xf domain. This process corresponds to perform a zoom in on specific intervals of the variables.

3.4. User authentication

A proposed model for monitoring app access events on mobile devices identifies abnormalities in access patterns of users. Access requests are denied if there are anomalies in access patterns determined by deviations in-app access events, called point anomaly classification. In the network administrator, a second authentication factor and next access request are rejected in this model. The second authentication factor in the model is provided by the user.

4. Experimental Results and Discussion

A description of the experimental results following the application of the fuzzy logic with the random forest method will be presented in this section. The proposed method is applied on a computer with 8GB RAM and 2.2 GHz running Python 3.7.3. In the weekday case, there are over 153,893 instances (events) and 65,487 during the weekend case. Additionally, there are approximately 1415 different apps that are described in this section. Using machine learning technique, compared the performance of the mobile application using the dataset as follows:

4.1. Dataset analysis

The 30 apps accessed by android mobile users on weekends and weekdays are used for this study. The weekends of 65,487 events out of 153,893 instances in a week are presently monitored in the dataset. During weekdays, the interaction time is approximately between 10.10 ms to 4.13 ms and on weekends, 19.04 ms to 278.99 ms of interaction time. This shows the difference in interaction for weekdays and weekends usage in-app. Compared to weekdays, the average access time is 44 % longer on weekends. The data analysis shows that average access per user is equal to other users during weekends and weekdays. The student statistical analysis t-test is used to measure statistical significant between weekday and weekend apps. A standard deviation of 9.044 in the weekday app and a standard deviation of 8.966 in the weekend app. The t value is calculated as 0.983, this is greater than 600 at p=0.05 that shows no significant difference between data samples.

4.2. Performance metrics

Various parameters are used to estimate the property of the proposed mobile application, along with comparisons with existing techniques. According to the proposed method, the following performance metrics Eq. (4-5) are considered:

$$Accuracy = \frac{Number\ of\ corre\ ct\ predictions}{overall\ predictions} \tag{2}$$

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

$$F - Measure = \frac{TP}{TP+1/2(FP+FN)} \tag{5}$$

4.3. Quantitative analysis

The proposed quantitative analysis of proposed fuzzy logic with the random forest method for mobile application is explained in this section. As shown in Table 1, values obtained using the proposed fuzzy logic algorithm combined with the random forest method for a mobile application are given. Based on the accuracy, precision, recall, and f measure evaluation results for the proposed fuzzy logic method with random forest for mobile application, the results are presented in Table 1.

Table 1: The quantitative analysis of proposed fuzzy logic with random forest method for mobile application

Metrics	Proposed Fuzzy logic with RF
Accuracy	98.67
Precision	94.32
Recall	95.54
F-score	99.98

Table 1 shows the quantitative analysis of proposed fuzzy logic with the random forest method for mobile applications usage. The proposed research method shows that it can perform better access control with authentication by monitoring the mobile apps usage on weekdays and weekend basis for better control of user over the mobile. The performance is evaluated in terms of accuracy, precision, recall, and f-score. An accuracy is some correct predictions to the overall predictions which are utilized to evaluate the model of classification. The proposed fuzzy logic with the random forest method for mobile application achieved an accuracy of 98.67%, precision of 94.32%, recall of 95.54% and f-score of 99.98%. The RF classifier is a simple process and constructs the subsets of example that classifies the original data correctly. The graphical representation of quantitative analysis of the proposed method is shown in Fig. 3.

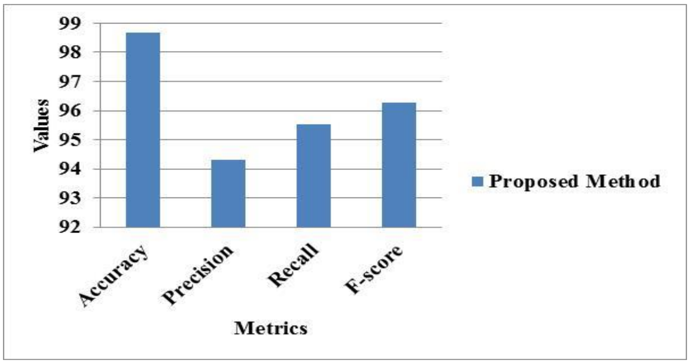

Fig. 3: The quantitative analysis graphical representation of proposed fuzzy logic with random forest method for mobile application

Table 2: The quantitative analysis of proposed fuzzy logic with random forest method for mobile application with the existing methods

Metrics	RNN	CNN	Fuzzy logic with DT	Proposed Fuzzy logic with RF
Accuracy (%)	89.56	90.23	93.45	98.67
Precision (%)	88.25	91.0	92.43	94.32
Recall (%)	90.23	91.25	92.39	95.54
F-score (%)	92.56	93.21	94.35	99.98

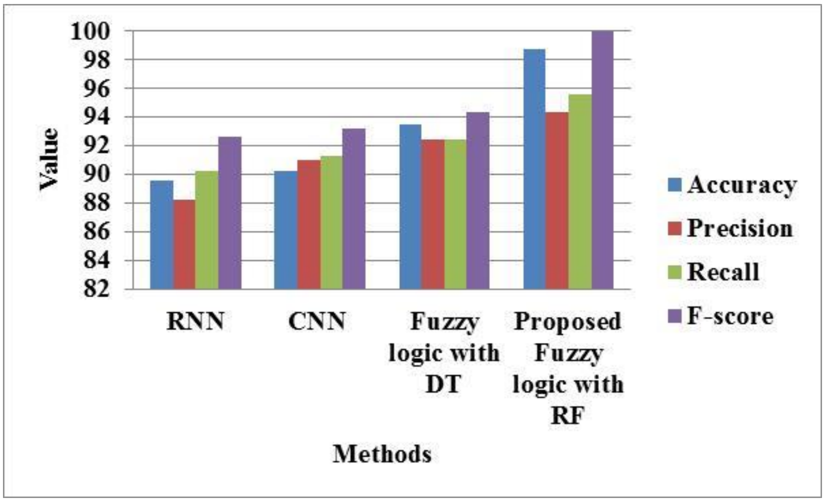

Fig. 4: The graphical representation of quantitative analysis of proposed fuzzy logic with random forest method for mobile application with the existing methods

Table 2 shows the quantitative analysis of proposed fuzzy logic with random forest method for mobile application with the existing methods. The performance is evaluated in terms of accuracy, precision, recall, and f-measure. The existing Recurrent Neural Network(RNN) method for mobile application showed an accuracy of 89.56%, precision of 88.25%, recall of 90.23% and f-score of 92.56%. The existing Convolutional Neural Network(CNN) method for mobile application showed an accuracy of 93.45%, precision of 92.43%, recall of 92.39% and f-score of 94.35%. The existing fuzzy logic with the Decision Tree (DT) method for mobile application showed accuracy of 90.25%, precision of 91.0%, recall of 91.25% and f-score of 93.21%. The proposed fuzzy logic with the random forest method for mobile application achieved an accuracy of 98.67%, precision of 94.32%, recall of 95.54%, and f-score of 99.98%. Fig. 4 shows a representation of the quantitative comparison of proposed and existing methods using the RF classifier, which is a simple process that produces the subsets of examples that classify the original data correctly. The findings show the proposed FuzzyRF method performs better in monitoring of mobile apps and provides accurate results on usage than the existing standard bench mark models.

4.4. Comparative analysis

Table 2 are presented the results of a comparative analysis of FuzzyRF for mobile authentication with the existing techniques such as Yosef Ashibani (2019) and Srivatsan (2020). According to Table 3, the proposed method and existing methods are compared.

Table 3: Comparison of proposed method and existing methods

Methods	F-Measure (%)
Yosef Ashibani (2019)	96.00
R. Srivatsan (2020)	99.29
Proposed Fuzzy logic with RF	99.98

In an analysis of performance metrics such as an F-measure, Table 2 compares the proposed method with existing methods. The proposed fuzzy logic with the random forest method for a mobile application is compared with existing methods such as Yosef Ashibani (2019) and Srivatsan (2020). This fuzzy logic with the RF method for mobile applications achieved an accuracy of 98.67%, precision of 94.32%, recall of 95.54%, and f-score of 99.98%. The existing method such as Yosef Ashibani (2019) and Srivatsan (2020) showed f-measure of 96.00% and 99.98% respectively. Therefore, the proposed method achieved higher performance compared with the existing method. The graphical representation of the comparison graph of proposed fuzzy logic with random forest method for mobile application is shown in Fig. 5.

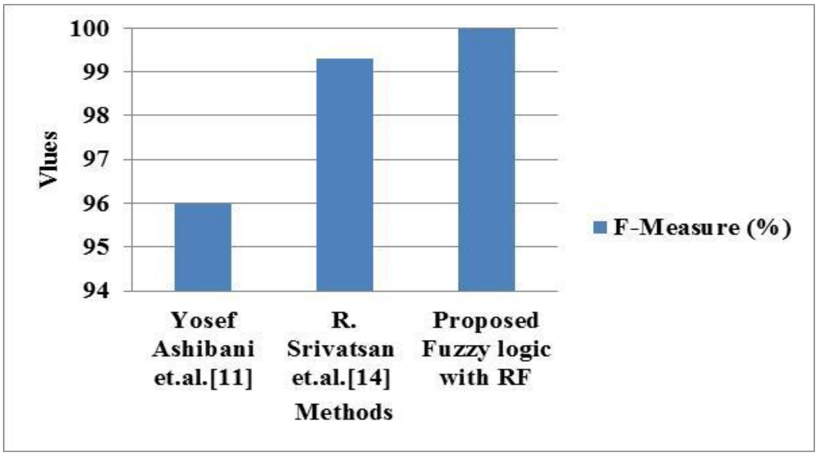

Fig. 5: The comparative analysis graphical representation of proposed fuzzy logic with random forest method for mobile application

5. Conclusion

The growing amount of smartphone applications and also the ability to install new apps on smartphones will make it possible to distinguish and verify the identity with a high degree of accuracy. This study presents a fuzzy-based RF based user identification approach in which app usage occurrences are analyzed. The proposed fuzzy basd RF method provides analysis on mobile app usage and user identification based on behaviour. Hence it provides better control for user on mobile usage addiction. Authentication and authorization events are used in a supervised learning user identification model for a technology system. The proposed method is tested with a real-world dataset. The model is tested for ongoing authenticated users who are using common apps at the same daily intervals throughout weekend days. The proposed model takes into account that users may operate their mobile apps separately during weekend days. The model is tested on a database and the findings show that the provided technique is capable of authenticating individuals with significant F-measure accuracy.

References

Amalfitano, D., Riccio, V., Amatucci, N., De Simone, V., & Fasolino, A. R. (2019). Combining automated gui exploration of android apps with capture and replay through machine learning. *Information and Software Technology*, 105, 95-116.

Arslan, R. S., Doğru, İ. A., & Barişçi, N. (2019). Permission-based malware detection system for android using machine learning techniques. *International Journal of Software Engineering and Knowledge Engineering*. 29(1), 43-61.

Ashibani, Y. & Mahmoud, Q. H. A machine learning-based user authentication model using mobile app data. In: *Proc. Of the International Conference on Intelligent and Fuzzy Systems*, 408-415.

García-Martín, E., Rodrigues, C. F., Riley, G., & Grahn, H. (2019). Estimation of energy consumption in machine learning. *Journal of Parallel and Distributed Computing*, 134, 75-88.

Gong, L., Li, Z., Qian, F., Zhang, Z., Chen, Q. A., Qian, Z., Lin, H., & Liu, Y. (2020). Experiences of landing machine learning onto market-scale mobile malware detection. In: *Proc. of the Fifteenth European Conference on Computer Systems*, 1-14.

Islamiati, D. S., Agata, D., & Besari, A. R. A. (2019). Design and implementation of various payment system for product transaction in mobile application. *In 2019 International Electronics Symposium (IES)*, 287-292.

Iphar, M. & Alpay, S. (2019). A mobile application based on multi-criteria decision-making methods for underground mining method selection. *International Journal of Mining, Reclamation and Environment*, 33(7), 480-504.

Junior, W., Oliveira, E., Santos, A., & Dias, K. A context-sensitive offloading system using machine-learning classification algorithms for mobile cloud environment. *Future Generation Computer Systems*, 90, 503-520.

Mattson, P., Reddi, V. J., Cheng, C., Coleman, C., Diamos, C., Kanter, G., Micikevicius, D., Patterson, D., Schmuelling, G., Tang, H., & Wei, G. Y. (2020). MLPerf: An industry standard benchmark suite for machine learning performance. *IEEE Micro*, 40(2), 8-16.

Shahidinejad, A. & Ghobaei-Arani, M. (2020). Joint computation offloading and resource provisioning for edge-cloud computing environment: A machine learning-based approach. *Software: Practice and Experience*, 50(12), 2212-2230.

Srivatsan, R., Indi, P. N., Agrahari, S., Menon, S., & Ashok, S. D. (2020). Machine learning based prognostic model and mobile application software platform for predicting infection susceptibility of COVID-19 using healthcare data. *Research on Biomedical Engineering,* 1-12.

Stanik, C., Haering, M., & Maalej, W. (2019). Classifying multilingual user feedback using traditional machine learning and deep learning. In: *Proc. Of the IEEE 27th International Requirements Engineering Conference Workshops (REW),* 220-226.

Thabtah, F. & Peebles, D. (2020). A new machine learning model based on induction of rules for autism detection. *Health Informatics Journal,* 26, 264-286.

Xu, X., Liu, Q., Luo, Y., Peng, K., Zhang, X., Meng, S., & Qi, L. (2019). A computation offloading method over big data for IoT-enabled cloud-edge computing. *Future Generation Computer Systems*, 95, 522-533.

Yeni, S., Cagiltay, K., & Karasu, N. (2020). Usability investigation of an educational mobile application for individuals with intellectual disabilities. *Universal Access in the Information Society*, 19(3), 619-632.