Science Arena Publications
Specialty Journal of Electronic and Computer Sciences
ISSN: 2412-7485
Available online at www.sciarena.com
2019, Vol, 5 (3): 75-85

An Efficient Method for Identifying Users across Various Digital Devices using the Modified XGboost Algorithm

Milad Ashrafi[1]*, Morteza Mohammadi Zanjireh[2]

[1]Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.
[2]Department of Computer Engineering, Imam Khomeini International University, Qazvin, Iran.

***Corresponding Author**

Abstract: *In the present study, the modified XGBoost algorithm is introduced for identifying users across different digital devices. Then, it is implemented on a dataset and the results are compared with the results of the decision tree, support-vector machine and K-nearest neighbor algorithms. Using several experiments with different parameters, it is proved that using 200 leaves and 1000 trees, the best result is achieved, which is the accuracy rate of 99.94% and the corresponding running time is 5439.26 seconds. The running time of the proposed algorithm is much greater as compared to the other algorithms tested in the present study, so, it shows poor performance in running time. But what makes this algorithm significant is its accuracy. According to the results, its accuracy rate is nearly 100% for any number of input data. The proposed algorithm shows that it can provide reliable results in the case of offline running where it does not matter how much its computing speed is.*

Keywords: *Decision tree, User Identification, Classification, Xgboost.*

INTRODUCTION

Today, an Internet user commonly has multiple devices connected to the Internet, such as computers and mobile phones. Although these devices are not directly interconnected, they show more or less the same common behaviors as they often belong to the same user. In addition, it should be noted that, as users utilize different devices, such as a computer, a tablet, a laptop, and a mobile phone, for doing their own Internet work, the User Identifier (UID) is fragmented, and this makes it more difficult to identify their identifiers (Wells, Fuerst and Palmer, 2005). Marketers need to be able to detect the time at which users are active on different devices, so that they can send recommendations, custom experiences, and meaningful messages in the form of email advertising or an offer to purchase from online stores of different websites.

Machine learning and data-driven methods have been processing in many areas, including the Internet. For example, machine learning is used for intelligent classification of spam emails. Below are some other applications of machine learning algorithm in the field of the Internet: the learning advertising systems are used to match proper advertisements with appropriate fields; fraud detection systems are used to protect the banking system against malicious attackers; deviant event detection systems are used to assist empirical physicists in identifying those events that lead to new physics. There are two important factors driving these

programs toward success: 1. To effectively use statistical models detecting the complex dependency of data; 2. To use scalable learning systems separating the targeted model from a large data set.
Among the machine learning methods practically used, the gradient boosting regression tree is a desirable method widely used in many applications (Friedman, 2001). By implementing a gradient boosting regression tree on many classification criteria in standard databases, significant results have been obtained (Li, 2010).
LambdaMAR is a type of tree used for ranking and significant results are obtained by using it for solving ranking problems (Burges, 2010). In addition, the LambdaMAR tree is also used as an independent predictor (He et al., 2014). It has been also used in the ensemble method and in challenges such as the Netflix Award (Bennett and Lanning, 2007).

Research background

Many studies have been carried out on the identification of users across various digital devices. For example, Carmagnola et al. (2009) provided a model in a form of conceptualization that provides a typical basis for cross-system personalization. Guna et al. (2014) have provided a gesture-based identification system suited for applications such as the user login to home multimedia services, with fewer security requirements. However, these methods are used to identify users based on specific information, and they don't consider the identification problem when users use different devices. Also, in (Hu, Gingrich and Sentosa, 2008), the k-nearest neighbor (KNN) algorithm was used to identify the user. The innovation of this method, as compared to the earlier nearest neighbor algorithm, is to consider different weights for different decisions (selection of different neighbors). In fact, it can be said that it is a cost-based user identification algorithm. In (Phoha et al., 2012), the Hidden Markov Model (HMM) was used to identify the user in a dynamic environment. Unlike most methods used for user identification, which are discriminative, the Hidden Markov Model is a generative model. Although this method has high accuracy, it needs a large amount of memory due to its generative nature. In addition, in (Sheng, Phoha and Rovnyak, 2005), a Parallel Decision Tree (PDT) algorithm was used to identify users. In this algorithm, users are identified by keystroke, although it has a relatively high false alarm.
The concept of the gradient (slope) boosting has been proposed by Friedman (2001). He proposed the TreeBoost model as well as a design method for training it. Then, Friedman (2002) introduced the box trick to boost the gradient. In addition, in order to solve the experiment speed problem, Friedman et al. (2000) have used the second-order statistics for tree splitting. Johnson and Zhang suggested a step-by-step correction and also the regulation of the tree complexity. However, the performance of these methods is not very good at running time and model size (Johnsonvand Zhang, 2013). The use of handheld smart devices, such as cell phones, has been widespread, and this increased use of these devices has raised the data security and privacy. The availability of handheld smart devices has made life easier. In (Ahmad et al., 2017), it is suggested that using a touch screen, an active user can log in the device through the protocol. In this way, the touch screen is considered as a canvas, known as an interaction trace map.
A modified XGBoost method is used to identify users across various digital devices. It is a scalable machine learning system. The impact of the system type has been widely recognized in a number of challenges to machine learning and data mining, for example, the challenges hosted by the Kaggle competition site. Of 29 challenging methods published in the Kaggle website in 2015, the XGBoost algorithm has been used in 17 methods. Among these methods, just eight methods have used XGBoost to train this model, while others have combined it with neural networks. For comparison, the second most popular method, i.e. deep neural networks, was used in 11 solutions. The success of XGBoost-based systems has also been seen in KDD Cup 2015, where XGBoost was used in top10 by any winning team. In addition, the winning teams reported that group methods (ensemble) outpace XGBoost a bit in performance (Ron Bekkerman, 2015).

These studies show that XGBoost provides very desirable results in solving a wide range of problems. Some example problems that can be solved using the XGBoost algorithm are store sales forecasting; high energy physics event classification; web content classification; customer behavior prediction; motion detection; click-through rate prediction; classification of destructors; product classification; risk prediction; MOOC (massive open online course) dropout prediction.

The most important factor for the success of XGBoost algorithm is its scalability in all scenarios. The system runs more than ten times faster than existing popular methods, and its scales are limited to billions of samples in distributed settings and memory settings. The scalability of XGBoost algorithm is due to several important systems and algorithm optimization. These innovations are including a new tree learning algorithm is used to handle sparse data; a weighted quantile sketch algorithm is used to handle weighted data effectively in tree learning; parallel and distributed computing accelerates learning, allowing a user to faster search a model. Above all, the XGBoost algorithm applies out-of-core computing, and enables data scientists to process hundreds of millions of samples on a desktop.

Problem statement

One of the common ways used to identify users is to rely on definite identifiers, such as names, email addresses, phone numbers, or other personal information (Setoguchi et al., 2014). However, marketers cannot use this personal information because it is confidential. That's why it's hard to identify users across various digital devices based on uncertain identifiers. A better solution for identifying users across different digital devices is to guess the device identifiers belonging to a user. To do this, it is required to use a classification method to identify and isolate different devices (Gueth et al., 2013). However, labeling sufficient training data for each device is highly costly and time-consuming. So, I'm interested in explaining whether it is technically possible to select some of the features to display the data, and whether it is time-consuming or not.

The present study aims to identify Internet users across various digital devices. Since users use different devices, such as laptops, tablets, cell phones, etc., during the day, to do their own Internet-based tasks, such as paying bills, checking email and university portal, online shopping, etc., it becomes more difficult and complex to identify their user ID, resulting in the need for new classification and user identification algorithms. Although in previous studies, some algorithms, such as decision tree, support vector machine, gradient boosting model and XGBoost, have been used to identify users, each one has been poor in the accuracy rate or computing time. Another objective of the present study is to modify and improve the efficiency of the XGBoost algorithm in order to identify the Internet users across different devices with a higher accuracy rate and computing speed (less computing time).

Also, the present study is our contribution to the digital marketing industry. Obviously, information on identification of user ID as well as the time at which users visit sites and also those sites more visited by the users can help internet marketers a lot. Indeed, by a better understanding of users and their interests, Internet marketers can offer more appropriate recommendations and suggestions in the form of emails and smart offers of online stores, and so on.

In the present study, a single-factor research environment is assumed. This means that just the XGboost algorithm and its parameters affect the research environment.

Moreover, the research environment is dynamic and variable. This means that the number of input and output data may vary at different times.

The proper selection of training data significantly affects the performance of the XGboost model.

In the present study, the variables are including: the number of nodes in each branch and tree depth in the XGboost decision tree and the number of samples in training data set and test data set. In this way, the final XGboost decision tree must have the optimal depth and width to reach the solution and it should be determined that what percentage of the samples should be placed in the training data set so that the system has higher accuracy and less error.

Method

In the present study, MATLAB software is used to evaluate and simulate the proposed method. The results are compared with other classification methods. Moreover, other classification error criteria, i.e. geometric mean and F-value, are investigated. These two criteria are used to investigate the effects of parameters used in one of the important steps of classification, namely clustering. Then, the results are compared with each other to symmetrically and asymmetrically grade and increase the number of intervals.

The proposed modified XGBoost algorithm has higher computing speed and accuracy than the original XGBoost model.

The proposed modified XGBoost algorithm has more applications in solving different issues, as compared to the original XGBoost model.

The proposed algorithm has higher computing speed and accuracy than the decision tree algorithm.

The proposed algorithm has higher computing speed and accuracy than the support vector machine algorithm.

The proposed algorithm has higher computing speed and accuracy than the k-nearest neighbor's algorithm.

XGboost algorithm

One of the famous gradient boosting algorithms is the lose-win decision in some Kaggle competitions. The XGBoost algorithm has a very high ability to predict and this makes it the best option for accuracy in various events because it has both a linear model and a tree learning algorithm. This algorithm performs almost 10 times faster than the gradient boosting algorithms. This algorithm contains various objective functions, regression, classification and ranking. One of the interesting points about the XGBoost algorithm is that it is also known as the Regulated Boosting Technique. This algorithm helps reduce large models and also has good support in a wide range of languages, such as Scala, Java, R, Python, Julia and C ++. This algorithm supports distributed training on different devices and includes GCE, AWS, Azure, and Yarn clusters. The XGBoost algorithm can be integrated into Spark, Flink, and other cloud data systems that are built based on cross-validation in each repetition of the boosting process.

The algorithm consists of the following steps: (Friedman, 2001)

1. $F_0(\mathbf{x}) = \arg\min_\rho \sum_{i=1}^{N} L(y_i, \rho)$
2. For $m = 1$ to M do:
3. $\tilde{y}_i = -\left[\frac{\partial L(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})}, \; i = 1, N$
4. $\mathbf{a}_m = \arg\min_{\mathbf{a}, \beta} \sum_{i=1}^{N} [\tilde{y}_i - \beta h(\mathbf{x}_i; \mathbf{a})]^2$
5. $\rho_m = \arg\min_\rho \sum_{i=1}^{N} L(y_i, F_{m-1}(\mathbf{x}_i) + \rho h(\mathbf{x}_i; \mathbf{a}_m))$
6. $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \rho_m h(\mathbf{x}; \mathbf{a}_m)$

Figure1. The regulated boosting algorithm

The algorithm used in the present study is as follows:

Algorithm 1 Multidimensional Output Boosted Regression Trees on Weighted Training Samples

1: $F_0(\mathbf{x}) = \bar{\mathbf{t}}$ ▷ weighted mean
2: **for** $m = 1$ to M **do**
3: $\tilde{\mathbf{t}}_i = \mathbf{t}_i - F_{m-1}(\mathbf{x}_i), i = 1, \ldots, N$
4: $(\mathcal{A}_m, \mathcal{R}_m) = \underset{\mathcal{A},\mathcal{R}}{\operatorname{argmin}} \sum_{i=1}^{N} w_i ||\tilde{\mathbf{t}}_i - H(\mathbf{x}_i; \mathcal{A}, \mathcal{R})||_2^2$
5: $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu H(\mathbf{x}; \mathcal{A}_m, \mathcal{R}_m)$
6: **end for**

Figure 2. The algorithm used in the present study

Results and Discussion

Data set

The data used in the present study are found in the CIKM Cup 2016 Track 2, which includes the CIKM Cup 2016, at the following link:
https://drive.google.com/drive/folders/0B7XZSACQf0KdXzZFS21DblRxQ3c
Or the link above is available at the following website:
http://cikm2016.cs.iupui.edu/cikm-cup/
The characteristics of the data set are listed in Table1:

Table 1. Characteristics of the data set

Number of device-logs	338,990
Number of events	66,808,490
Average/Median number of events per device-log	197 / 106
Number of URLs	14,148,535
Number of URts having title	1,796,480
Number of unique words in titles	8,485,859

(b) Statistics of datasct partitions

Statistics	**Training**	**Validation**	**Test**
Number of device-logs	240,732	50,146	48,112
Number of users	60,001	12,528	11,993
Number of golden links	506,136	107,654	107,653

Preprocessing operations

To use the data set, the following operations are applied:

- To separate the first 20,000 records from the data set, due to the limited work space and RAM. The total number of records is more than 800 thousand, which requires a supercomputer or a distributed network for processing.
- To convert selected records to a two-dimensional array of integers
- To convert the last column of each record, according to being TRUE or FALSE, to 1 or zero as the output of each record.

Practical experiments

To perform a practical experiment, two main parameters of the algorithm, the number of leaves and the number of trees, are investigated. It is observed that as the number of trees increases, running time increases linearly, indicating that increasing the number of trees results in increased running time with a constant gradient (Fig.3).

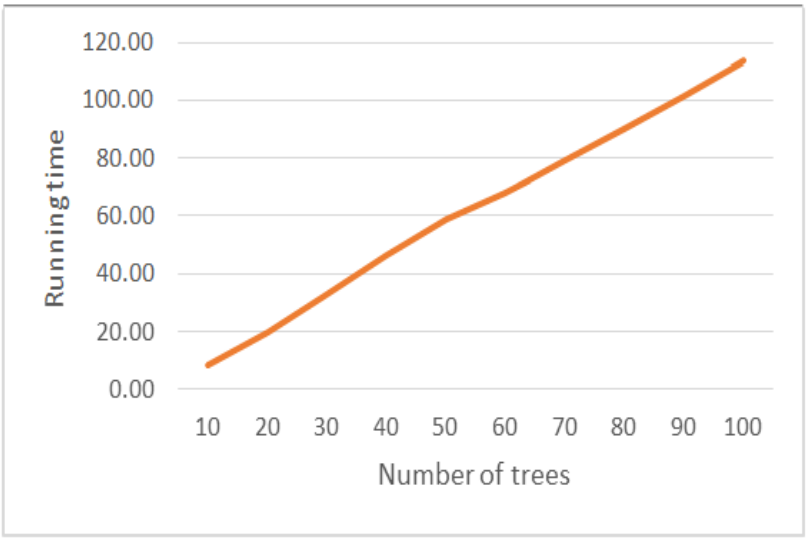

Figure 3: Variation of running time against the number of tree

Also, as the number of trees increases, the correct detection rate first increases (here up to 80 trees), and then decreases (Fig.4).

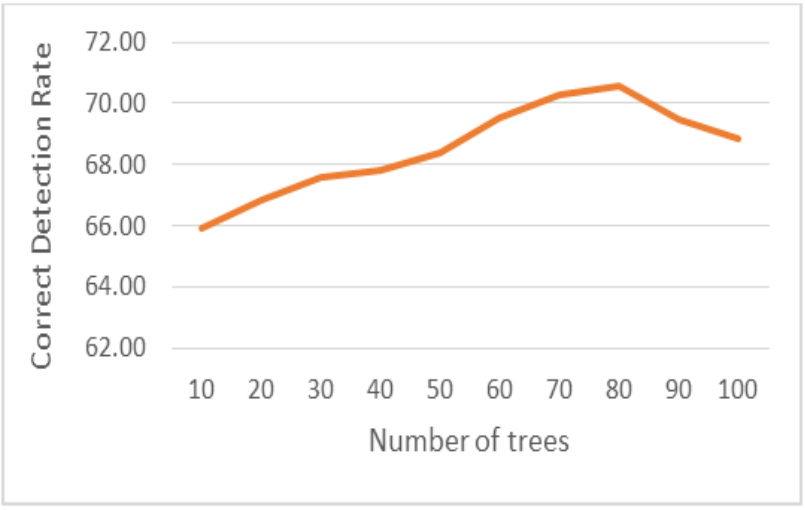

Figure 4: Variation of correct detection rate against the number of trees

In the above experiments, the number of leaves parameter was considered constant and equal to 10 leaves. For a better comparison, the experiment was performed with 200 trees and the correct detection rate was obtained as 65.4, which was less than that obtained with 100 trees. With 300 trees, it was obtained as 61.58, showing a decreasing trend in the correct detection rate.

As the number of leaves parameter increases, as expected, the gradient becomes steeper. The reason for this is that the leave calculation is performed in all the trees, so the running time of the proposed algorithm greatly depends on the number of leaves (Fig. 5).

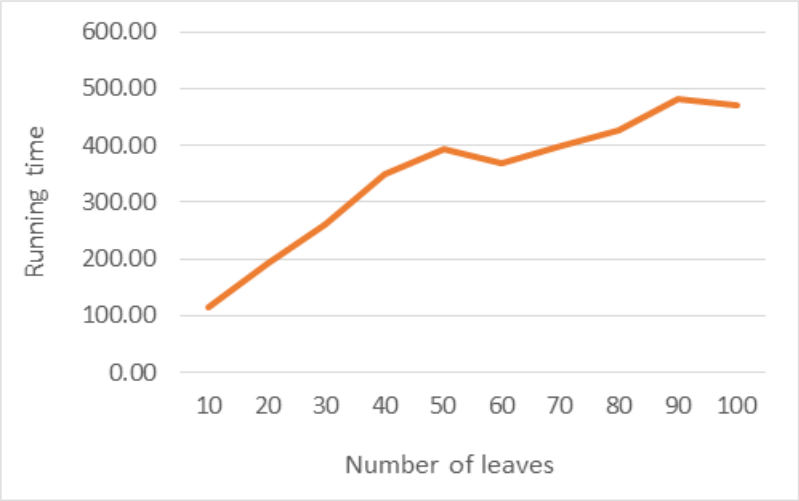

Figure 5: Variation of running time against the number of leaves

As the number of leaves increases, the correct detection rate varies and significantly changes at a given number of leaves, which is 70 leaves here. This means that, firstly, the number of leaves does not affect the correct detection rate; secondly, after a given number of leaves, the number of trees should not be increased depending on the number of leaves. This is discussed further below. (Fig.6)

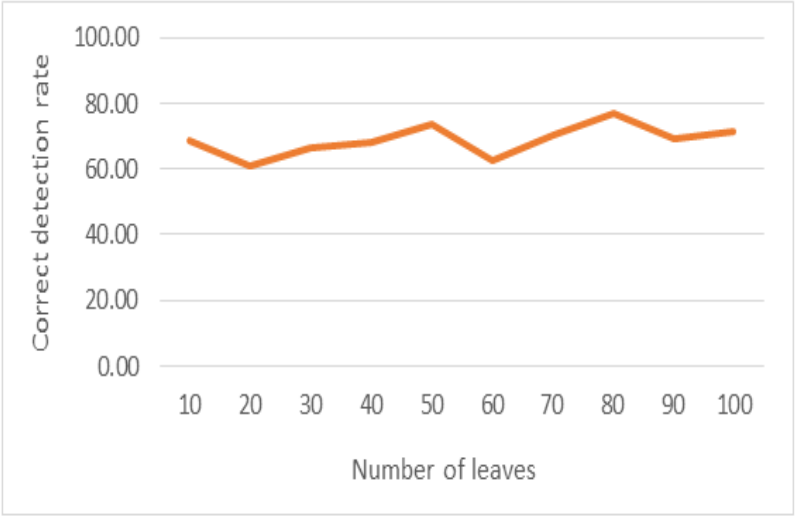

Figure 6: Variation of correct detection rate against the number of leaves

So far, the effect of the two main parameters of the algorithm is investigated. According to the results of several experiments with different numbers of trees and leaves, it is concluded that 200 leaves and 1000 trees will provide the best result, which is the accuracy rate equal to 99.94% and the corresponding running time is 5439.26 seconds.

To better investigate the performance of the proposed method, it was compared with three other decision tree, SVM and KNN algorithms. In the running time, the proposed method really shows poor performance. The need for further calculations would increase the running time. This difference is shown in the figure below.

Figure 7: Comparison of the four studied algorithms in running time

As shown in the figure above, the running time of the proposed algorithm (in blue line) was so higher as the running time of the other three algorithms seem to be almost linear. The lowest running time is related to the KNN algorithm, as shown in the figure below.

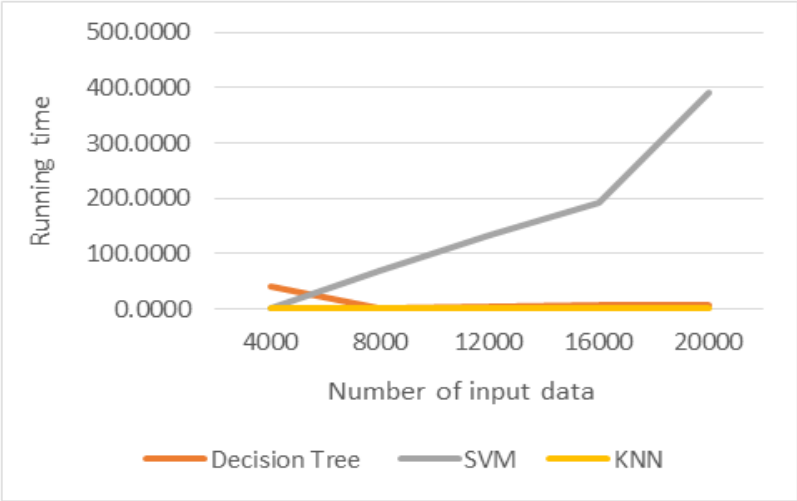

Figure 8: Comparison of the three decision tree, SVM and KNN algorithms in running time

As seen in the figure above, the running times of KNN and the decision tree are close, but SVM requires much time for processing because of the input dimension. To classify 8 attributes, SVM requires nearly one-million-time repetition. That is why its running time is longer but still less than the proposed method. The strength of the proposed algorithm is its accuracy rate.

Figure 9: Comparison of the four studied algorithms in correct detection rate (or accuracy rate)

About the accuracy rate, the result obtained for the proposed algorithm is amazing. It is observed that the accuracy rate of the proposed algorithm per any number of input data is nearly 100%. So, it is concluded that it outperforms the other three algorithms in the accuracy rate, followed by SVM, KNN and decision tree, respectively.

Conclusion

In the present study, the modified XGBoost method was introduced aa an optimal solution for identifying users across different digital devices. Then, it was implemented on a data set. In the practical experiment section, after preprocessing the data, the effects of the two main parameters of the algorithm, i.e. the number of leaves and the number of trees, on the performance of the algorithm were investigated. It was observed that as the number of trees increased, the running time almost linearly increased, indicating that an increased number of trees results in the increased running time with a constant slope.
But, as the number of leaves increases, the slope becomes steeper, as expected. The reason for this is that the leaves calculation is performed in all the trees, so the running time greatly depends on the number of leaves.
With various experiments performed with different parameters, it was found that it is concluded that 200 leaves and 1000 trees will provide the best result, which is the accuracy rate of 99.94% and the corresponding running time is 5439.26 seconds which is a reliable value. The results of the experiments are listed in the table below.

Table 2: Results of the practical experiments described in the research

Accuracy	Time	Tree Num	Leaf Num	Accuracy	Time	Tree Num	Leaf Num
65.92	8.39	10	10	68.82	113.54	100	10
66.81	20.05	20	10	60.89	191.02	100	20
67.60	32.96	30	10	66.59	262.09	100	30
67.82	46.36	40	10	68.19	349.03	100	40
68.41	58.90	50	10	73.60	393.48	100	50
69.53	67.88	60	10	62.52	368.12	100	60
70.28	78.99	70	10	70.17	398.31	100	70
70.55	90.13	80	10	76.71	427.37	100	80
69.50	101.16	90	10	69.42	481.66	100	90
68.82	113.54	100	10	71.20	471.02	100	100

65.41	227.76	200	10	86.54	624.26	100	150
61.59	345.62	300	10	99.94	5439.26	1000	200

To compare the proposed algorithm with the three decision tree, SVM and KNN algorithms in the running time, the results of experiments are listed in the table below.

Table 3: The comparison of the proposed algorithm with the decision tree, SVM and KNN algorithms in the running time

Number of input data	Proposed algorithm	Decision tree	SVM	KNN
4000	707.6403	41.6517	2.7599	0.7104
8000	1456.5000	2.8114	68.6293	0.3542
12000	3024.8000	4.6436	132.8951	0.8208
16000	3920.0000	6.3705	192.6197	1.9046
20000	5439.2551	7.2415	391.8165	3.0004

As stated, the proposed algorithm operates worse than other three algorithms in the running time. But, what makes this algorithm more significant than other three algorithms is its accuracy rate (correct detection rate), as shown in the following table as compared to the other three algorithms.

Table 4: The comparison of the proposed algorithm with the decision tree, SVM and KNN algorithms in the accuracy rate (correct detection rate)

Number of input data	Proposed algorithm	Decision tree	SVM	KNN
4000	100.0000	44.4000	68.2000	56.8000
8000	99.9990	48.3500	56.6500	54.1500
12000	100.0000	44.0000	58.5300	55.5700
16000	99.9255	43.7500	62.1000	56.1700
20000	99.9400	44.8600	63.8200	55.5400

According to the results of the present study, the accuracy rate of the proposed algorithm is nearly 100% per any number of input data. The proposed algorithm shows that it can provide reliable results in the case of running offline where it does not matter how much its computing speed is.

References

1. Ahmad, J., Sajjad, M., Jan, Z., Mehmood, I., Rho, S., & Baik, S. W. (2017). Analysis of interaction trace maps for active authentication on smart devices. Multimedia Tools and Applications, 76(3), 4069-4087
2. Burges, C. J. (2010). From ranknet to lambdarank to lambdamart: An overview. Learning, 11(23-581), 81.
3. Carmagnola, F., & Cena, F. (2009). User identification for cross-system personalisation. Information Sciences, 179(1-2), 16-32.
4. Friedman J., Hastie T., R. T.-T. annals of statistics, and undefined 2000, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," projecteuclid.org.
5. Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189-1232.
6. Gueth, P., Dauvergne, D., Freud, N., Létang, J. M., Ray, C., Testa, E., & Sarrut, D. (2013). Machine learning-based patient specific prompt-gamma dose monitoring in proton therapy. Physics in Medicine & Biology, 58(13), 4563.

7. Guna, J., Stojmenova, E., Lugmayr, A., Humar, I., & Pogačnik, M. (2014). User identification approach based on simple gestures. Multimedia tools and applications, 71(1), 179-194.
8. He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., ... & Candela, J. Q. (2014, August). Practical lessons from predicting clicks on ads at facebook. In Proceedings of the Eighth International Workshop on Data Mining for Online Advertising (pp. 1-9). ACM.
9. J. F.-C. S. & D. Analysis and U. 2002, "Stochastic gradient boosting," Elsevier, 2002.
10. Johnson, R., & Zhang, T. (2013). Learning nonlinear functions using regularized greedy forest. IEEE transactions on pattern analysis and machine intelligence, 36(5), 942-954.
11. Li, P. (2010). An empirical evaluation of four algorithms for multi-class classification: Mart, abc-mart, robust logitboost, and abc-logitboost. arXiv preprint arXiv:1001.1020.
12. Phoha, V. V., Phoha, S., Ray, A., Joshi, S. S., & Vuyyuru, S. K. (2012). "Hidden markov model ("HMM")-based user authentication using keystroke dynamics." U.S. Patent 8,136,154, issued March 13.
13. Ron Bekkerman, (2015). "The Present and the Future of the KDD Cup Competition," [Online]. Available: https://www.kdnuggets.com/2015/08/kdd-cup-present-future.html. [Accessed: 12-Feb-2018].
14. Setoguchi, S., Zhu, Y., Jalbert, J. J., Williams, L. A., & Chen, C. Y. (2014). Validity of deterministic record linkage using multiple indirect personal identifiers: linking a large registry to claims data. Circulation: Cardiovascular Quality and Outcomes, 7(3), 475-480.
15. Sheng, Y., Phoha, V. V., & Rovnyak, S. M. (2005). A parallel decision tree-based method for user authentication based on keystroke patterns. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 35(4), 826-833.
16. Wells, J. D., Fuerst, W. L., & Palmer, J. W. (2005). Designing consumer interfaces for experiential tasks: an empirical investigation. European Journal of Information Systems, 14(3), 273-287.
17. Bennett, J., & Lanning, S. (2007, August). The netflix prize. In Proceedings of KDD cup and workshop (Vol. 2007, p. 35).
18. Hu, J., Gingrich, D., & Sentosa, A. (2008, May). A k-nearest neighbor approach for user authentication through biometric keystroke dynamics. In 2008 IEEE International Conference on Communications (pp. 1556-1560). IEEE.