International Journal of Research in Advent Technology, Special Issue, ICATESM 2019
E-ISSN: 2321-9637
Available online at www.ijrat.org

Dynamic Authentication By 'Risk Level Analysis Based User Profiling' Using ML Approach

Pratik Ingawale[1], Parth Khaladkar[2], Omkar Mhatre[3], Aishwarya Kusagal[4]
Department of Computer Engineering,
PVG's College of Engineering and Technology,
VidyaNagari, Maharashtra, Pune-India - 411009.
Email:pratikingawale6@gmail.com [1], khaladkar.parth@gmail.com [2], omkar_mhatre12@yahoo.com [3], kusagalaishwarya@gmail.com [4]

***Abstract*—**In today's world there are many cases of hacking and security breaches of online accounts such as social networking and web accounts. In order to prevent this, we propose a Risk Based Authentication System using Machine Learning. This system will work in three phases. First phase is fetching and preparing historical user login data indicating user profile and User Parameter Extraction. Phase two of the Risk Based Authentication System is Risk Engine. The input to this phase is the parameters and historical data obtained as output from phase one. This phase consists three ML models working simultaneously on same input - SVM Classifier, One-Class SVM Classifier and Bayesian Classifier. Bayesian Classifier and SVM calculate the probability of user being fraud by analyzing user profile and current login data while one-class SVM gives Risk Score. Phase three of the Risk Based Authentication System is recommendation of Authentication Methods. This phase decides the complexity of the authentication that user will face during login based on the risk level. Using risk score and probabilities of SVM and Naïve-Bayes user is assigned a risk level among four risk levels. Different authentication techniques are graphical password, OTP, email verification, security questions. The combination of these techniques is assigned to each risk level used for increasing the complexity of login according to the different scenarios.

Keywords— Machine Learning Algorithms, Multi - factor Authentication, Naïve-Bayes' Classifier, One-class SVM, Support Vector Machine (SVM), Risk-Based Authentication, Risk Engine.

1. INTRODUCTION

In this digital world, communication via internet has become an integral part of our life. As it has become a necessity, securing communication is a major challenge. Hacking into accounts and accessing private information has become a major concern. So correctly authenticating user before allowing him/her the access rights is important. Username and password are the most commonly used authentication techniques used to validate a user into an application or online system. This is knowledge-based approach and so is easily vulnerable to threats. Hence such static single factor authorization does not ensure complete security. Moreover, the device does not do any computation to verify whether the user is genuine or not. Biometrics on the other hand are expensive to implement. Login parameters and credentials used in a combination can provide better security. A user login record is analyzed using eight user parameters which are collected during current login attempt for a specific user. A user is profiled according to his historical patterns and classified as genuine or fradulent before logging in using login credentials and attributes. These attributes are analyzed using Machine Learning model consisting of three trained modules through which risk level is calculated and used to determine the authentication techniques in different combinations for a particular user. Use of three algorithms increases accuracy of classification. So, a fradulent user will have to go through multiple randomly generated set of authentication techniques while a genuine one will not. This ensures that security and usability of system is maintained. This also provides dynamic nature to the authentication system as the user is unable to predict the authentication technique. This concept promises a cost efficient and low computational authentication system.

2. HISTORY & BACKGROUND

This section looks into the similar attempts and models that have been reported to check for discrepancies in the user login patterns and help address breach issues. The first and the most popular one is the Google Two-Step Authentication [1] which is implemented to sign into google accounts consisting of Gmail, Drive etc. This is an optional parameter that can be enabled by the user which in order to prevent compromising your account even though your password has been stolen, asks you for code that will be sent to your phone via text/voice call. Thus with 2-step verification even if someone hacks through your password layer, they will still need your phone to get into your account [2]; but, this as mentioned needs to be enabled and being static does not involve the device doing any computations to classify the user. Also, stolen devices can lead to security issues.

The above stated system thus does not use machine learning to implement dynamic authentication. The proposed models help classify the users as genuine or fradulent based on machine learnt profiling of the user login patterns. Both types of Machine learning algorithms are used to profile different user login parameters and create a user profile that can be used for providing complex login methodologies.

A Risk-Based Authentication System was proposed by M Misbahuddin, B Bindhumadhava [3]. A risk engine model based on three ML algorithms that are SVM, Naïve Bayes and one class SVM is used to calculate risk level associated with user login by comparing current login record with risk profile. The Risk profile is obtained through past login records of the user. Then, as per risk level user is challenged with different login method corresponding to each risk level. However, **only one** method per risk level is used which is **fixed,** making system vulnerable if this single method at each risk level is somehow compromised.

This drawback has been eliminated in our approach which uses three login methods per risk level. These multiple level of authentication methods for each risk level are decided dynamically at runtime (that is any two methods out of available three pertaining to each of the risk level are selected at runtime) thus ensuring perfect tradeoff between usability and security.

3. DESIGN ISSUES

The dynamic authentication system consists of three modules Data generation, Risk engine and Authentication.

3.1 Data Generation Module:

The data generation module includes the following steps to prepare relevant dataset from raw dataset.

3.1.1 *Creation of account*:

User has to create an account by providing valid data such as Email Id, mobile number, pattern lock, answers to security questions, graphical password. This is the primary registration step that is required to accept the credentials which will later be used for validating the entry of user.

3.1.2 *Parameter extraction:*

When the user enters the username, validation mechanism at server-side checks whether user account exists or not. If yes then real time login parameters such as OS, Browser, IP, Device, Time Zone, Login time, Geo location, Number of failed attempts are extracted from the session.

3.1.3 *Data preprocessing:*

a. As the input records consists of attribute LoginTime which depends on Timezone. To resolve this dependency of these two parameters we introduce a third parameter called GlobalStandardTime which converts the login time into standard format according to (G.S. T). Then the Timezone and LoginTime are discarded and further computations are carried out by considering login time according to global standard time.

b. Encode the data as all ML algorithms require numeric data as input.

3.1.4 *Fetch user history:*

As per the username the past login records for particular user has been fetched from database for further analysis. The fetched history and extracted parameters from current login are passed to Risk Engine module for further analysis and computation of risk scores in Risk engine module.

3.2 Risk Engine:

The risk engine module contains the main business logic. It works on the user dataset and compute the probability and risk score. It has the three main machine learning algorithms

3.2.1 *Support Vector Machine (SVM):*

For classification and regression problems, Supervised Machine Learning can be used.SVM is a supervised Machine Learning Algorithm which uses labelled data. SVMs find a hyperplane that best divides a dataset into multiple classes.[4]

3.2.2 *One Class SVM:*

One-class SVM classification, as its name implies is a unary classification or class-modelling [5] which aims at identifying objects belonging to a specific class amongst all objects inputted, by learning from a training set containing only the objects of that class. It is different and difficult than traditional methods that with the training set containing objects from all the classes distinguish between two or more classes.

One-class SVM is concerned with only single class known as novelty/normal class. In the context of this paper, this class is "Genuine" class of users logging in signifying that a genuine user is logging in. One-class SVM is unsupervised ML algorithm makes use of unlabeled Data. There are no rules for learning provided explicitly. One-class SVM finds

consistent patterns among input data samples which all belong to the same class by itself.

In context of this paper, training dataset for one-class SVM uses Genuine User Login Patterns only. The output obtained is a prediction whether a test user login record is genuine (TRUE) or not (FALSE).

UserLogin Parameter	Weight
Browser	1
OS	2
LoginTime	3
IPAddress	4
Device	5
NoOfFailedAttempts	6
GeoLocation	7
TimeZone	8

Table 1. User Parameters and respective weights

With the help of above table which assigns different weights to different user parameters based on their importance, risk score for a user is calculated. The login parameters in current user login record are checked for historic references of values. If current record has a user parameter with value which is not in the historic login data of same user, risk score is incremented by the risk weight associated with that parameter as specified in Table 1: Weight Table

Formula for calculating risk score is:

$$RiskScore = \sum(UserParameterValue * UserParameterWeight)$$

…Eq. (1)

where,

$$UserParameterValue = \begin{cases} 0, & if\ behaviour\ exists\ in\ history \\ 1, & Otherwise \end{cases}$$

3.2.3 *Naïve-Bayes classifier:*

Naïve-Bayes Classifier is a machine learning algorithm for classification problems. It is based on Bayes' probability theorem. Primarily it is used for text classification which involves high dimensional training data sets. Naive-Bayes' uses conditional probability and probability of one event provided another event happens.

Naive-Bayes assumes statistical independence of data features, that is, neither direct or inverse dependencies between data.

Algorithm:

Bayes theorem [6] provides a way of calculating the posterior probability, P(c|x), from P(c), P(x), and P(x|c).

Naive Bayes classifier assume that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors.

This assumption is called class conditional independence.

$$P(c|x) = \frac{P(x|c) * P(c)}{P(x)}$$

$$P(c|x) = P(X1|c) * P(X2|c) * P(X3|c) *\ldots. P(Xn|c) * P(C)$$

…Eq. (2)

- P(c|x) is the posterior probability of class (target) given predictor (attribute). Here it is probability of user being fraudulent(c) provided user record(x)appears
- P(c) is the prior probability of class. That is, probability of user belonging to class fraudulent.
- P(x|c) is the likelihood which is the probability of predictor given class. This indicates probability of user record x existing in records with fraudulent class records. x= $\{x_1,x_2,\ldots x_n\}$; x_1 to x_n are attributes of user record x. Here, the eight attributes are Browser, OS, IPAddress, GeoLocation, Device, LoginTime, TimeZone, NumberOfFailedAttempts.

The advantages of Naive-Bayes' Classifier [7]:

- It works well with minimum training data.
- It can handle missing feature values by Laplace smoothening and also other smoothing techniques. Hence, it can work with data not having all features.
- Naive-Bayes' Classifier ensures regularization of the output. The regularization is achieved inherently as probability values lie between 0 and 1.
- Naive-Bayes' Classifier is easy to implement and computationally efficient as compared to other

Machine Learning algorithms as it involves simple probability calculations.

3.3 Authentication module:

The authentication module with the help of the risk score by one-class SVM and probabilities of being fraud by SVM and Naive-Bayes' classifier computes the risk level, randomly presents the user with the authentication methods. The user thus gets classified according to the risk level he is assigned into. One to Four signifies risk threat or probability of the user being fradulent from minimum to maximum.

Risk level calculated by risk engine is given to this module as an input. Risk level is mapped with different authentication methods as per Table 2.

The user will be presented by any two or three of methods randomly from the set presented for each risk level as mentioned in the table given above.

If user successfully passes through all the proposed authentication methods, only then access is granted

The System Architecture indicating three phases is given in Fig. 1

The flow chart of given system is depicted in Fig. 2

Table 2. Determining Risk Level and Corresponding Login Method

SVM or Bayesian Probability(S)	One-Class SVM Risk Score (R)	Risk Level	Authentication Methods
$0.5 < S \leq 0.6$	$1 \leq R \leq 6$	1	Security Question, Password, Email-Verification
$0.6 < S \leq 0.75$	$7 \leq R \leq 18$	2	Password, Email-Verification, OTP
$0.75 < S \leq 0.9$	$19 \leq R \leq 29$	3	Email-Verification, OTP, Pattern Lock
$0.9 < S \leq 1$	$30 \leq R \leq 36$	4	OTP, Graphical Password, Pattern Lock

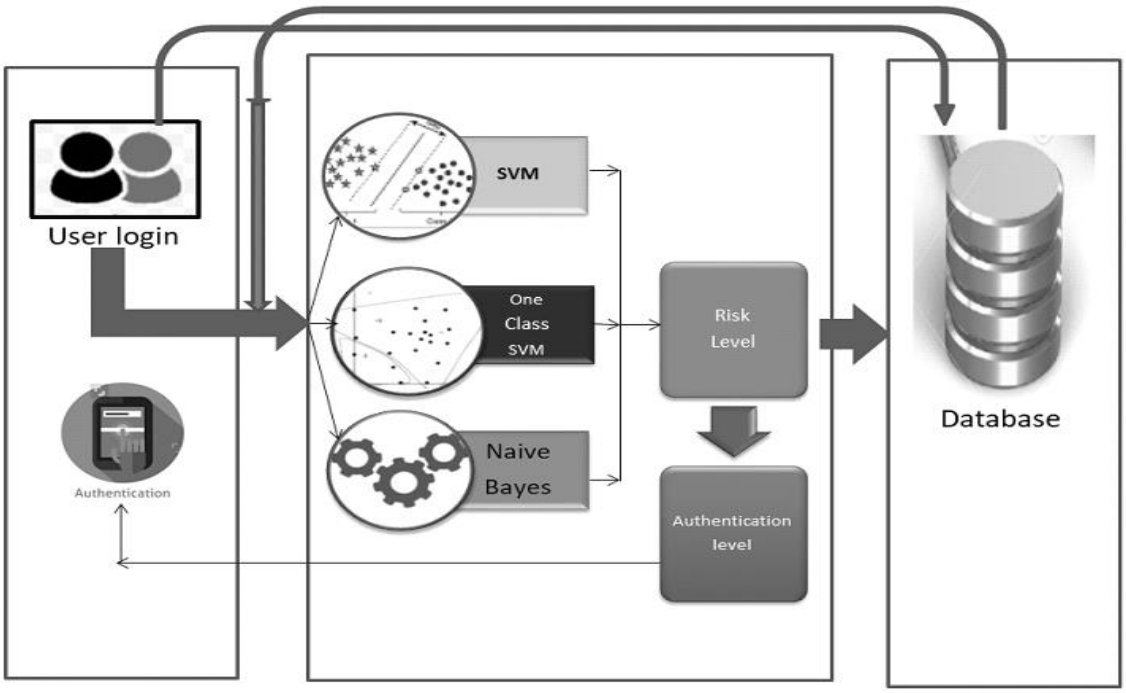
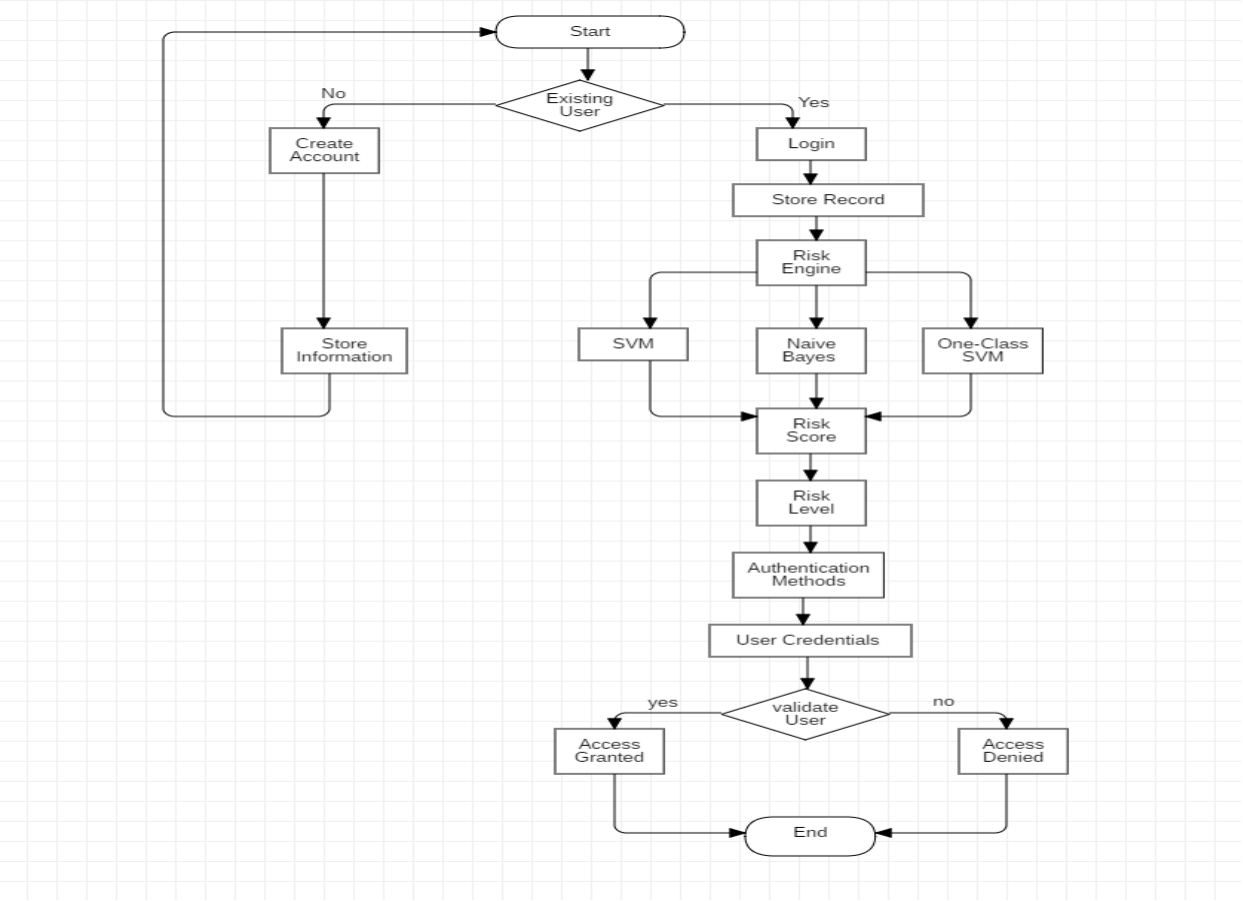


Fig. 1 System Architecture



. Fig. 2 Flowchart of System Control

4. RESULT AND ANALYSIS

The proposed method is implemented using Python and Django framework. During the first login attempt after account creation, user ID and password are used for authentication as no previous records are available for ML models to make predictions on. It is not before 10 records are accumulated in the database for a particular user that the Risk Engine is activated. After this minimum requirement is satisfied models learn user profile through behaviour implied by past login records. The constraints applied in this model are:

(1) Number of failed attempts impacts the Risk Level only after two unsuccessful attempts.
(2) One-hour deviation from usual login time in historical records is considered as risk. The usual login time is calculated as that time value (only hour value will be considered) which has occurred most frequently.

Genuine user patterns as illustrated in Table 3 are used form training One class SVM.According to the user records in Table 3, the user behavior that can be established is that the user frequently logs in at different times of the day from location Pune, using Windows OS.

SVM and Naive Bayes classifier being supervised algorithms require both genuine and fraudulent patterns. The fraudulent patterns for the same user with userID 1 are shown in Table 4. The features of fraudulent patterns can be observed are changes in few of the key attributes like geo-location, IP address and timezone. Owning to the difficult authentication methods as a result of elevated risk level that these users could not pass, these records were classified as fraudulent by one class SVM.

The different user records with different user parameters trying to login in a user account with userID 1 are shown in Table 5 after a behavioural profile has been established. These records indicate performance of risk engine in different scenarios. As can be inferred from Table 5 the first two scenarios being are classified identical to the learned user profile and so give less probabilities of being fraudulent with low risk scores and probabilities, since not much change is

being observed from regular genuine patterns. In third scenario, change in device, OS, Browser, IPAddress and number of failed attempts higher than threshold affect output considerably. But as location and timezone have not changed the outputs have medium range values. In the fourth scenario different TimeZone, GeoLocation, Device, OS, Browser and large number of failed attempts does have severe impact on the classification probabilities and risk score, which have shot up for all three algorithms indicating fraud user trying to break into account. Thus, based on user profiling we can classify the user into Genuine and fraudulent and present him with complex authentication techniques.

Table 3. Genuine Login Patterns

userID	IPAddress	GeoLocation	TimeZone	LoginTime	OS	Browser	Device	#FailedAttempts	Class
1	192.25.25.225	Pune	05:30:00	04:40:00	Windows 10	Firefox	HP	0	Genuine
1	192.25.25.225	Pune	05:30:00	13:58:00	Windows 10	Firefox	HP	0	Genuine
1	192.25.25.225	Pune	05:30:00	04:25:00	Windows 10	Firefox	HP	0	Genuine
1	192.25.25.225	Pune	05:30:00	08:20:00	Windows 10	Firefox	HP	0	Genuine
1	192.25.25.225	Pune	05:30:00	17:28:00	Windows 10	Firefox	HP	0	Genuine

Table 4: Fraudulent Login Patterns

userID	IPAddress	GeoLocation	TimeZone	LoginTime	OS	Browser	Device	#FailedAttempts	Class
1	103.33.21.41	Pune	05:30:00	15:52:00	Windows 8	Chrome	Acer	1	Fraudulent
1	108.16.67.98	Bangalore	05:30:00	19:05:59	Ubuntu	Firefox	Dell	4	Fraudulent
1	109.24.68.78	Delhi	05:30:00	21:14:59	Mac	Safari	Apple	2	Fraudulent
1	18.23.10.40	Cupertino	16:00:00	00:18:00	Windows 7	Firefox	Dell	2	Fraudulent
1	182.31.29.6	NewYork	19:00:00	21:07:59	Ubuntu	Firefox	HP	0	Fraudulent

Table 5: User Login Scenario Representation

Scenario	One	Two	Three	Four
User ID	1	1	1	1
IP Address	111.41.23.2	192.25.25.225	112.247.3.4	122.68.92.1
TimeZone	05:30:00	05:30:00	05:30:00	16:00:00
Login Time	04:20:23	10:30:32	14:20:34	18:41:55
OS	Windows	Windows	Mac	Ubuntu
Browser	Firefox	FireFox	Safari	Chrome
Device	HP	HP	Apple	Lenovo
GeoLocation	Mumbai	Pune	Pune	Cupertino
Failed Attempts	1	0	3	4

SVM	0.5323	0.2346	0.7865	0.9187
Naive Bayes	0.21	0	0.8923	0.9786
One Class SVM	11	3	18	31
Risk Level	2	1	3	4

5. CONCLUSION

Authentication is the primary entry point for any system and is often neglected by many providers. Dynamic Authentication By 'Risk Level Analysis Based User Profiling' Using Machine Learning Approach makes use of different parameters to first classify the behaviour of the user whether suspicious or not, based on analysis of collected historical data thus making system more realistic, as ML algorithms extract patterns or rules inherently that characterise genuine or fraud activity implied by past activity. Accordingly, it provides security measure combinations to user, thus demanding suitable additional verification based on user profile in order to provide better security.

Any variation in the normal pattern in treated as suspicious and used to generate a risk level. The system makes use of SVM, Naive Bayes and One class SVM thus handling both labelled and unlabelled data making it useful for variety of data as well as these three algorithms work together to cover each other's shortcomings, making overall output more relevant and correct.

This system thus proposes a secure method to protect online accounts from authentication threats.

6. FUTURE SCOPE

The proposed system is related to the security and ML domain. As per the requirement of the application the authentication techniques can be scaled up. Other authentication techniques like biometrics and digital signature can be used. The algorithms used are standard machine learning algorithms. The efficiency and accuracy of these algorithms become saturated after certain level and cannot be increased beyond it. Thus, to increase the efficiency use of Deep Learning can be incorporated.

Machine Learning models can become better progressively but may still need some refinements, that is, if machine learning models make inaccurate predictions, the machine learning engineer has to step in and may require to reapply different techniques of feature engineering and data preprocessing and retrain the model[8]; but, if used deep learning algorithms can learn the inherent and implicit feature relations and patterns within data on their own without human assistance, solely through training data[9] making system robust even further.

REFERENCES

[1] https://www.google.com/landing/2step/, Google Two Step Verification
[2] https://www.google.com/landing/2step/#tab=how-it-protects , "Google 2-step Verification - How it protects?"
[3] M Misbahuddin, B Bindhumadhava, B Dheeptha, 'Design of a Risk-Based Authentication Techniques Using Machine Learning Approach', IEEE 2017
[4] https://en.wikipedia.org/wiki/Support-vector_machine , SVM Wikipedia
[5] http://rvlasveld.github.io/blog/2013/07/12/introduction-to-one-class-support-vector-machines/ , One-class SVM GitHub.io
[6] https://en.wikipedia.org/wiki/Naive_Bayes_classifier , Naïve-Bayes Classifier
[7] https://courses.edx.org/courses/course-v1:Microsoft+DAT275x+2T2018/course/ , Principles of Machine Learning-Python Edition, edx.org
[8] Giuseppe Bonaccorso, "Machine Learning Algorithms", Packt Publishing Limited**,** ISBN-10: 1785889621, ISBN-13: 978-1785889622, Feature Selection and Data Preprocessing
[9] https://en.wikipedia.org/wiki/Deep_learning#Deep_neural_networks , Deep Learning, Wikipedia Article