International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

Quick Random MAC Address Detection Based on Organizationally Unique Identifier in Captive Portal

Imam Riadi[1], Abdul Fadlil[2], Basit Adhi Prabowo[3]

[1] *Department of Information System, Universitas Ahmad Dahlan, Yogyakarta, Indonesia*
[2] *Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia*
[3] *Master Program of Informatics, Universitas Ahmad Dahlan, Yogyakarta, Indonesia*

*E-mail address: imam.riadi@is.uad.ac.id, ** fadlil@mti.uad.ac.id, *** 2208048011@webmail.uad.ac.id*

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: Privacy has become a big concern for both countries and individuals when using the internet. Many countries and standards committees have created regulations or guidelines addressing these privacy issues. Some companies adapt by implementing random MAC addresses. Recently, some operating systems have made random MAC addresses the default option instead of device MAC addresses. Random MAC address detection is necessary due to problems arising in certain scenarios in captive portals. This research proposes a MAC address classification formula with two threshold variables. Data was taken from the database of devices that successfully logged in to the captive portal. The class whether random or not is determined by the Organizationally Unique Identifier part of the given MAC address of the device. It was challenged with Gaussian Naïve Bayes, Logistic Regression, K-nearest neighbors, and New Support Vector Classification to get the threshold value with the highest accuracy and F1-score. These threshold values are used to replace the variables in the classification formula. The results of the classifiers provide the same accuracy pattern, with accuracy values between 93.7993% and 98.1139%, and F1-score values between 93.8424% and 98.1342%. Gaussian Naïve Bayes produces the optimum both accuracy and F1-score. Random MAC address detection can be implemented in a captive portal.

Keywords: Privacy, Random Media Access Control, Captive Portal, Supervised Machine Learning, Organizationally Unique Identifier

1. INTRODUCTION

The International Telecommunication Union (ITU) releases data on individuals who use the Internet around the world, either based on surveys conducted by each country or ITU estimates [1]. In 2022, the proportion of persons who use the internet in the world is 66.3%. Some years ago, in 2005, the proportion was 15.6%. In ten years until 2015, there has only been an average 2.44% rise in internet use. Early in the COVID-19 pandemic's outbreak from 2019 to 2020, the usage of the internet, either via fixed or cellular networks, increased by 5.9%. During the COVID-19 pandemic's later years, from 2020 to 2022, the increase was an average of 3.35% per year.

The exceptional nature of the Internet and digital platforms's rapid growth has given rise to various issues concerning consumer privacy since the 1990s [2]. Consumer behavior is no longer collected locally and anonymously. As a response to this, operating system makers embedded a feature to create random MAC addresses in their products, initiated by Apple in 2014 [3]. In the upcoming year, Google (2015) and Microsoft (2016) implemented it in their operating systems.

IEEE released a document practice that specifies a privacy threat model for IEEE 802® technologies in 2020 [4]. Additionally, the document offers suggestions for safeguarding against issues related to privacy. Device fingerprints can be obtained from permanently assigned MAC addresses. These fingerprints can threaten user's privacy. A device fingerprint is a relationship between a device and observable information elements, especially related to user identification. OUI as a part of a MAC Address can lead to information about the person's wealth and a person's location that can be located by information from GPS with the persistent fingerprint. Using randomly generated MAC addresses is one method to alleviate this privacy risk [5]. Random addresses therefore make it harder to identify a user device through its MAC/L2

E-mail:author's email

address. Applying random addresses may cause problems in some settings, such as captive portal (AAA) and DHCP [6]. The MAC address is used in AAA as a part of device authorization. When a MAC address is changed, device data transmission on the port is closed until the AAA server verifies that the new MAC address is verified.

In the captive portal, every device that successfully logs in will occupy the slot provided to the user for some number of MAC addresses and information related to the connection will be recorded. Some of them use enabled by default random MAC addresses to leverage privacy to another level. The device will change its MAC address each time it starts a new connection on the next day. This causes it to use another slot available in a long device timeout setting, even though it is the same device that was authorized the day before.

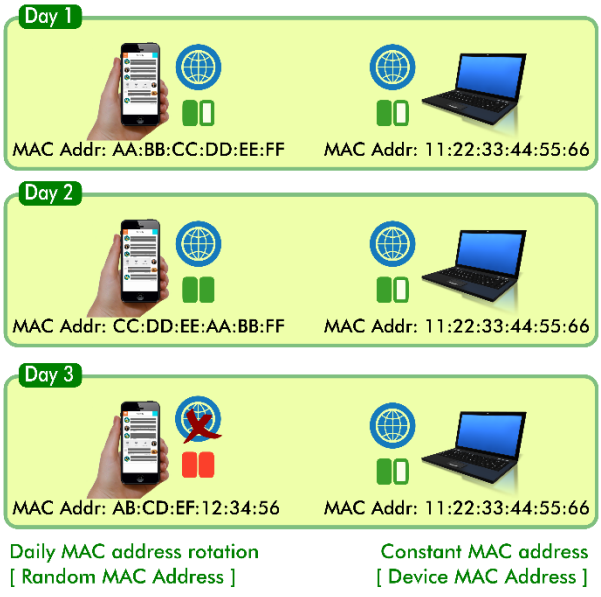


Figure 1. Random MAC address versus device MAC address behavior

If all slots have been used, the device will no longer be able to connect to the internet. Therefore, to provide further treatment, a method for detecting the presence of random MAC addresses is required.

2. Related Work

Research has been conducted in the past to perform binary classification using several classifiers. Detection of DDOS attack research [7] in 2020 with relatively high precision using Artificial Neural Networks (ANN), Support Vector Machine (SVM), Gaussian Naïve Bayes, Decision Tree (entropy-gini), and Random Forest. High precision DDoS classification, **closest to 100%** using K-nearest neighbors (KNN), Logistic Regression, and Naïve Bayes (Multinomial – Bernoulli algorithms). Another DDOS attack detection research [8] in 2023 using Logistic Regression resulted in low accuracy. On the other side, K-nearest neighbor (KNN) and Random Forest were **nearly 100%**. Research [9] in 2017 used average (μ) and standard deviation (δ) to classify DDoS attacks with a Gaussian Naïve Bayes classifier. The normal class area with μ+(3δ) and the attack class area with μ+(2,5δ) obtained an **accuracy of 100%**. Email spam detection with high precision and low time complexity, performed by research [10] in 2021 with an accuracy of **almost 100%** in Logistic Regression, outperforms Deep Learning, Naïve Bayes, and Neural Network.

Some researchers are concerned with MAC address randomization. Research [11] in 2021 on random MAC address detection on Bluetooth Low Energy (BLE). This research uses the signal strength emitted by the device to the access point to associate the device with a random MAC address. Research [12] in 2022 on the existence of random MAC addresses on Wi-Fi networks. This research proposes to use one SSID with multiple passwords or multiple random SSIDs to associate devices with MAC addresses. The use of random MAC addresses on 160 device models made in the period 2012 to 2020 was researched in [13] in 2021. This research uses OUI in some cases to correlate MAC addresses with the manufacturer or operating system. According to research [13], the main lesson learned is that randomization technology has not been applied consistently or cleanly to the variety of modern mobile devices. Different manufacturers adopting the same operating system have their unique features and distinctions, in addition to the fact that different OSes introduce these technologies in different ways.

Different from research [13], this research used OUI to find a threshold value in a formula that can be used to help decide quickly whether a device uses a random address or not. The formula will be tested with Logistic Regression (LR) [7],[10], Gaussian Naïve Bayes (GNB) [9], K-nearest neighbor (KNN) [7], [8], and New Support Vector Classification (nuSVC) rather than another classifier. This research does not associate the MAC address with the device as research [11] and [12].

3. Research Methods

This research proposes a quick way to identify random MAC address usage of devices connected to a captive portal based on the Organizationally Unique Identifier (OUI) [5] of the device's MAC address expressed in (1)

$$y_i = \begin{cases} 1, if \left(\frac{\sum_j loginnum_j < M}{|MAC\_Address|}\right)_i \geq P \; and \; \mu login_i < M; \\ 0, \qquad otherwise. \end{cases} \quad (1)$$

where i = OUI group iterator; j = data source iterator per OUI; y_i = classification of MAC address based on OUI; $loginnum_j$ = login counter per user per MAC address; |MAC_Address| = cardinality of MAC Address [14]; P = threshold for the percentage of random MAC address; μ $login_i$ = login average based on OUI; and M = threshold for device MAC address identification;

OUI is taken from the first 6 hexadecimal digits out of 12 hexadecimal digits [5], or 24 bits out of 48 bits of the device's MAC address. The steps to obtain a classification formula that can be used to detect random MAC addresses quickly are shown in Fig. 2.

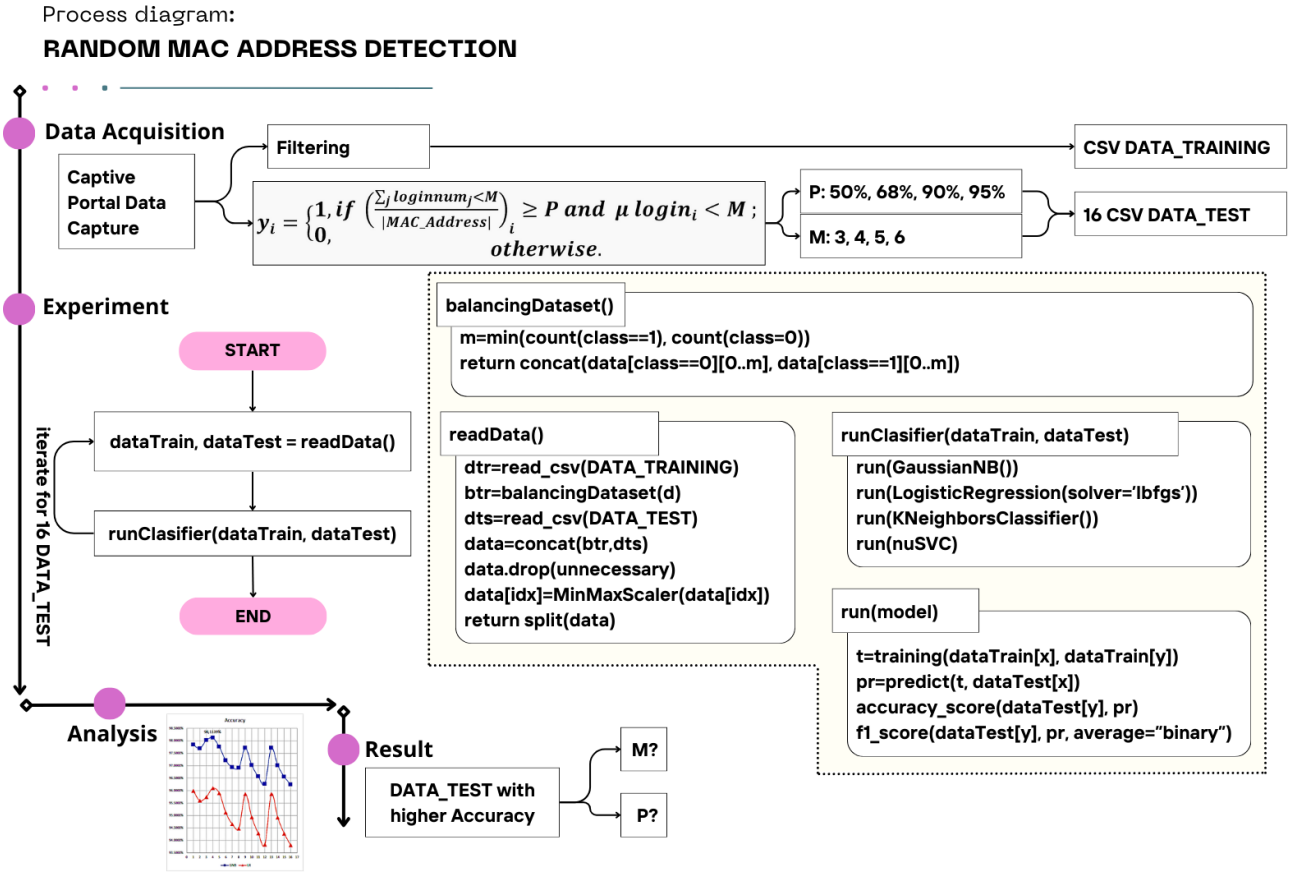


Figure 2. Process diagram

A user was trying to connect to the internet using the device through the network. The traffic was intercepted by the captive portal, then examine the ID in the database. If the data indicate that the ID was valid and satisfies all the requirements required, then the device was allowed to continue the connection.

Every successful user login data was stored in the database. The data was filtered out to get balanced training data and the rest was used as 16 different test data using a formula with variables of M and P. The unnecessary data column was removed and not used in the iteration. Both training data and test data are kept in the range of 0 to 1 so that all data outside that will be treated with MinMaxScaler.

All of the data was challenged with four different supervised machine learning algorithms: LR, GNB, KNN, and nuSVC. Prediction result compared with the initial class of each data row to calculate accuracy and F1-score.

The values of the variables M and P were obtained from the highest accuracy and F1 scores among the 16 test data challenged. The final values of M and P were fed back into the formula to get a new formula that can be used to label the new data.

4. Data Acquisition

A. *Data Capture*

Research data was obtained from users who successfully logged in to the captive portal. Captive portal use a database to store accounting data and some configuration data [15]. The aggregate of the current successfully logged-in device count from the built-in authentication table—for example, radpostauth—was inserted into the new radrandomisefactor custom table. This process was proposed in this research and expressed algebraically in (2):

$$\begin{aligned} &insert\ into\ radrandomisefactor \\ &\pi_{username,callingstationid,COUNT\ (1)\rightarrow count} \\ &\gamma_{username,callingstationid,COUNT\ (1)} \\ &\sigma_{reply="Access-Accept"}\ (radpostauth) \end{aligned} \qquad (2)$$

where callingstationid = MAC Address of user's device

Formula (2) was converted to a trigger of the radpostauth table on row update expressed in (3):

$$if\ reply = \text{Access-Accept}, then$$
$$insert\ into\ radrandmisefactor$$
$$(username, callingstationid)$$
$$VALUES\ (NEW.username, NEW.callingstationid)$$
$$ON\ DUPLICATE\ KEY\ UPDATE$$
$$loginnum = loginnum + 1 \quad (3)$$

Both (2) and (3) produce information about the number of devices used by a certain user associated with a specific MAC address. A MAC address represents a single device owned by a user.

B. The Creation of Training Data

The recap data in the radrandomisefactor table was filtered with some criteria to create training data inserted into the custom table `research.validation` group by OUI of the device. Similar criteria applied to training data of the device MAC address class as well as the random MAC address class: at least two distinct users utilized the same OUI (criteria 1). A requirement for the device's MAC address class was that all devices (100%) had logged in more than eight times (criteria 2). The random MAC address class was based on the rerement that all devices (100%) have logged in fewer than five times (criteria 3). All of the criteria to create training data are expressed in algebraic (4)

$$insert\ into\ research.validation$$
$$\pi_{oui,0 \rightarrow pctrand,AVG\ (\text{loginnum}) \rightarrow loginnumavg,0 \rightarrow israndomclass}$$
$$\sigma_{COUNT\ (1) \geq 2\ AND\ COUNT\ (1) = SUM\ (\text{loginnum} \geq 8)}$$
$$\gamma_{oui,COUNT\ (1),SUM\ (\text{loginnum} \geq 8),AVG\ (\text{loginnum})}$$
$$(radrandomisemacfactor)$$
$$\cup$$
$$\pi_{oui,1 \rightarrow pctrand,AVG\ (\text{loginnum}) \rightarrow loginnumavg,1 \rightarrow israndomclass}$$
$$\sigma_{COUNT\ (1) \geq 2\ AND\ COUNT\ (1) = SUM\ (\text{loginnum} < 6)}$$
$$\gamma_{oui,COUNT\ (1),SUM\ (\text{loginnum} < 6),AVG\ (\text{loginnum})}$$
$$(radrandomisemacfactor) \quad (4)$$

where criteria 1: COUNT(1) >= 2; criteria 2: COUNT(1) = SUM(loginnum>=8); and criteria 3: COUNT(1) = SUM(loginnum<6)

The results of (3) were then saved into a DATA_TRAINING.csv file. Training data creation is illustrated in Fig. 3.

#	User_Id	Mac Address	n Login
1	123456	AB:CD:EF:12:34:56	9
2	654321	AB:CD:EF:98:76:54	11
3	112233	EF:CD:AB:56:78:90	1
4	123456	AA:BB:CC:34:56:78	1
5	776324	AA:BB:CC:98:AE:BD	1
6	482634	AA:BB:CC:A1:B2:C3	1
7	654321	CD:EF:AB:AA:BB:CC	10
8	729213	CD:EF:AB:12:34:56	8
9	729213	CD:EF:AB:42:54:1A	7

OUI	n User	Class
AB:CD:EF	2	Device Mac
AA:BB:CC	3	Random Mac

criteria 1:
min. 2 distinct
users of overall

criteria 2:
100% login >= 8 times
criteria 3:
100% login <6 times

notes:
the OUI EF:CD:AB on the fourth row did not match criteria 1
the OUI CD:EF:AB on the 7th-9th row did not match criteria 3 (2/3 login >= 8 times)

Figure 3. Illustration of training data creation

All user's device login data were recorded in the database. This data was then aggregated to obtain information on the number of logins that have been carried out, grouped by the user ID and OUI of the device. From the example above, there are 4 OUI groups, namely AB:CD:EF (2 users, 2 devices), EF:CD:AB (1 user, 1 device), AA:BB:CC (3 users, 3 devices) and CD:EF:AB (2 users, 3 devices). OUI AB:CD:EF met criteria 1 and criteria 2: 100% logged in more than or equal to 8 times. OUI EF:CD:AB did not meet criterion 1. OUI AA:BB:CC met criterion 1 and criterion 3: 100% logged in less than 6 times. OUI CD:EF:AB did not meet criterion 1, because CD:EF:AB:42:54:1A did not meet criterion 2, while the other 2 devices met criterion 2, so they did not meet the 100% criteria.

C. Generating Test Data

Test data was generated by applying the value of M=[3,4,5,6] and the value of P=[50%,68%,90%,95%] or P=[0.5, 0.68, 0.9, 0.95] to (1) with data from radrandomisefactor table exclude training data stored in `research.validation` table, expressed algebraically in (5):

$$\pi_{oui,SUM(loginnum<M)/COUNT(1)\rightarrow pctrand,\ AVG(loginnum)<M\rightarrow loginnumavg,\ SUM(loginnum<M)/COUNT(1)\geq P\ AND\ AVG\ (loginnum)<M\rightarrow israndomclass}$$
$$\gamma_{oui,SUM\ (loginnum<M),COUNT\ (1),AVG\ (loginnum)}$$
$$\sigma_{NOT\ (voui=\pi_{voui}(research.validation))}$$
$$(radrandomisemacfactor) \quad (5)$$

Formula (5)'s results were stored in 16 DATA_TEST_M_P.csv in CSV data format with the following sequence as shown in Table I.

TABLE I. RESEARCH AND TEST DATA GENERATING SEQUENCE NUMBER

		P			
		50%	***68%***	***90%***	***95%***
M	***3***	1	5	9	13
	4	2	6	10	14
	5	3	7	11	15
	6	4	8	12	16

D. Data Privacy Protection

Metadata related to user privacy in this research is the username and MAC address. The dataset contains metadata that can be used for other research but is not used in machine learning. This research uses OUI, but it is not closely related to user privacy.

Privacy-related metadata is masked with general-purpose hash because they are consistent in producing results. Consistent result is needed because the data is used in filtering and grouping. To avoid dictionary attacks, salt is used [16], as in (6)

$$maskedword \leftarrow hash(original_{word} + salt, algo) \quad (6)$$

It is possible to use different salt on each column, but this research uses only one salt for every data masked on each column.

Password hashes are not used because they have inconsistent results. It makes it impossible to group the same real username or the MAC address. The comparison of the two types of hashes and how password hash generates different results each called is shown in Table II.

TABLE II. HASH COMPARISON USING GENERAL-PURPOSE HASH AND PASSWORD HASH

original_word (username)	**salt**	**masked_word**			
		general purpose hash		**password hash**	
		md5 #1	**md5 #2**	**bcrypt #1**	**bcrypt #2**
abc	900150983cd24fb0d6963f7d28e17f72	81515170fcfa683c8c9542529233515b	81515170fcfa683c8c9542529233515b	$2y$10$2aE429RgBOBhOKTgTiPP7.abFzw.phPeW/lcDLansxjGh6q9fd9Wm	$2y$10$XBKFbs36le6v1u8JqnNfbeffJufdvoUdApKeP9xlSie.J3WbEcyx.
123		c0ce0eea53f2f97e22f7ec247a698c88	c0ce0eea53f2f97e22f7ec247a698c88	$2y$10$zGhi5X83HfpNnz2U8q.Cb.3pI4MU8/RrqP0kS7wK2b.ZW3nvVQDcu	$2y$10$ggJq.BFLUVKBQRnBKbOj5euSrmSUdjklppBL6drXa0rfMcw3GY/gG

5. EXPERIMENT

A. Read Training Data

Training data obtained from DATA_TRAINING.csv was stored in the DataTrain variable. The quantity of training data for each class must be equivalent to prevent bias when performing classification tests. For the smaller size of classification, all of the data was picked, while for the largest one, data was picked randomly to make an equivalent size of the dataset as shown in Fig. 2 section balancingDataset().

B. Read Training Data

Test data read from each DATA_TEST_M_P.csv file in an iterative manner was saved in the DataTest variable. Dimensions were remained unaltered for both classes.

C. Data Pre-processing

Both training data and test data have the same metadata. Unnecessary columns were discarded from the dataset variable, if any. Pctrand, loginnumavg, and israndomclass columns were kept. Each piece of data was maintained in the range of 0 to 1 as shown in Fig. 4. MinMaxScaler was applied to out-of-range data.

Figure 4. Illustration MinMaxScaler

The minimum value is 5 and the maximum value is 25. All values must be maintained in the range 0 and 1 with a scale of 1:(25-5)=1:20.

The pctrand and israndomclass columns were already in the range of 0 and 1, so these columns did not need to be processed. Meanwhile, the loginnumavg column needs to

be processed with MinMaxScaler because the maximum value is more than 1.

The final step of data pre-processing was separating the classification from the dataset. A dataset is separated into a multidimensional list of parameter data and a one-dimensional list of classification data. There were four lists available for the classification challenge: DataTraining[parameter], DataTraining[class], DataTest[parameter], and DataTest[class] as shown in Fig. 5.

pctRand	LoginNumAvg	isRandomClass	#
1	0.00473	1	trainingdata
1	0.00540	1	trainingdata
1	0.00867	1	trainingdata
1	0.00946	1	trainingdata
0	0.37116	0	trainingdata
0	0.38771	0	trainingdata
0	0.68794	0	trainingdata
0	1.00000	0	trainingdata
0.6941	0.21320	0	testdata
0.7037	0.22770	0	testdata
0.8095	0.12200	1	testdata
0.1875	0.96730	0	testdata
0.1667	0.64570	0	testdata
0.7647	0.17330	1	testdata
0.7	0.31060	0	testdata
0.5625	0.18610	1	testdata
0.3846	1.00000	0	testdata
0.4632	0.59220	0	testdata

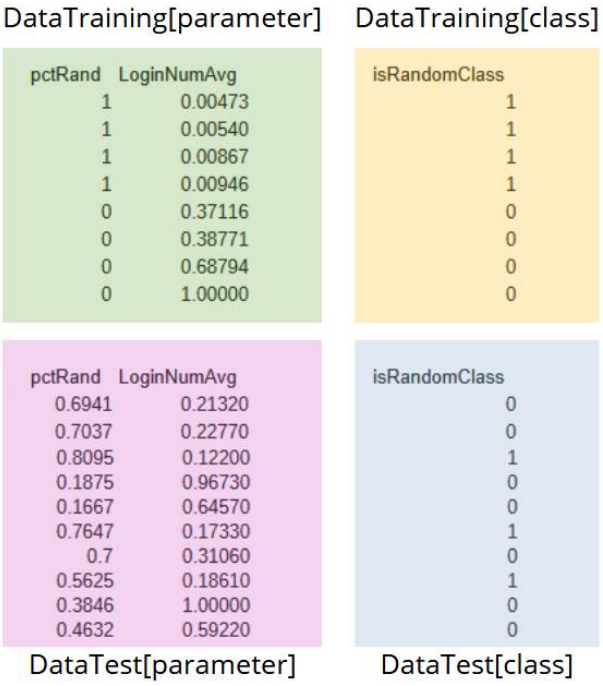


Figure 5. Data pre-processing results

D. Classification Challenge

The algorithms of GNB and LR differ, but the procedures followed in machine learning to find M and P values in (1) were identical: training, predicting, calculating the accuracy, and calculating the F1-score.

The Bayes theorem is a technique for creating an updating process for probability depending on new information. The posterior probability, or P(A|B), indicates the likelihood that A will occur given that B has occurred. The use of Naive Bayes on normally distributed data is known as Gaussian Naive Bayes. For every x_i inside y_k, Gaussian Naive Bayes assumes that the likelihood $P(x_i|y)$ follows the Gaussian Distribution, as indicated by (7) [9]

$$P(x_i|y) = \frac{1}{\delta\sqrt{2\pi}}\, e^{-\frac{1}{2}\left(\frac{x_i-\mu}{\delta}\right)^2} \qquad (7)$$

where μ = average; e = Euler constant (2.71828); π = 3.14159; δ = standard deviation; and x = independent variable;

Logistic Regression is one of the methods for modeling the connection between predictor (independent) variables and target (dependent) variables. This is a widely used method for categorical data classification and prediction. This method uses the logit function (log-odds) to relate the predictor variable to the likelihood that the target variable will occur expressed in (8)[17], producing output in the form of a restricted probability between 0 and 1.

$$p(Y=1) = \frac{1}{1+e^{-logit(p)}} = \frac{1}{1+e^{-(\propto + \sum \beta_i X_i)}} \qquad (8)$$

where p(Y=1) = probability of success event (class 1 or Y = 1); logit(p) = relationship between independent variable and probability of success; α, β = coefficients estimated during the model training process; and X = independent variable;

KNN is a straightforward classifier whose effectiveness is determined by the K value chosen and the similarity metrics (distance functions) applied. The K value indicates how many elements are neighboring the target. If the target is close to a large number of elements of a class, then KNN will predict the target as a member of that class [18].

NuSVC [19] is slightly different from SVC. NuSVC uses a new parameter ν instead of C to fine-tune control over the model's complexity and generalization capabilities, where ν is in the interval between 0 and 1. This parameter serves as an upper bound on the fraction of margin errors and a lower bound on the fraction of SVs.

Training refers to the process of teaching a machine learning model to make predictions or decisions based on input data. The model learns from the training data by adjusting its parameters or internal weights to minimize the

difference between its predictions and the actual outcomes, resulting in accurate predictions on new data. DataTraining[parameter] and DataTraining[class] were used in this process.

Predict refers to the process of using a trained model to make predictions or generate outputs for new data. In this procedure, the model obtained from training and DataTest[parameter] was used.

The accuracy value and F1 score value show the effectiveness of the classification model expressed in (9) and (10):

$$A = \frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

$$F1 = \frac{TP}{TP+\frac{(FP+FN)}{2}} \quad (10)$$

where A = Accuracy; F1 = F1-score; TP = row counts of true positive criteria; TN = row counts of true negative criteria; FP = row counts of false positive criteria; and FN = row counts of false negative criteria;

The accuracy value is used to measures the proportion of correct predictions and F1-score value is to ensure that both precision and recall are sufficiently high.

Using the F1-score along with accuracy provides a more comprehensive evaluation of a classification model's performance. There is a chance that high accuracy fails to reflect the model's poor performance in identifying the minority class. It happens for the combination of high accuracy and low F1-score.

6. Result and Discussion

A. Data Acquisition

Formula (2) or (3) produced 26,138 recapitulations of user data recorded by the radius server in the database. It was saved in the radrandomisefactor table.

Formula (4) results in 611 rows of training data. Of the 611 data rows, 107 were device MAC address classes, and the remaining 504 were random MAC address classes.

Various M and P values applied to (5) produced 16 distinct test data. Each test data consists of 15,583 rows grouped by OUI and saved in a different file.

B. Load and Preprocessing Data

Each test data in the DATA_TEST_M_P.csv file was loaded into the DataTest variable. The size of each DataTest variable was 15,583 rows.

Load training data from DATA_TRAINING.csv into the DataTrain variable. Of the 504 random MAC addresses in DataTrain, 107 were picked randomly. 107 rows of random MAC address and 107 rows of device MAC address data joined together to replace the current DataTrain.

DataTest and DataTrain join together to do pre-processing. In the pre-processing stage, all columns were discarded from DataTrain, but for the pctrand, loginnumavg, and israndomclass columns. MinMaxScaler was applied for the loginnumavg column to update the value in the range of 0 and 1. The results were split into four lists. DataTrain[parameter] and DataTest[parameter] contain pctrand and loginnumavg columns. DataTrain[class] and DataTest[class] contains an israndomclass column.

C. Iteration For Each Test Data

The iteration was carried out according to the sequence in Table I. DataTrain[parameter] and DataTrain[class] were trained with a classifier producing a classification model. The model was used to make classification predictions for DataTest[parameter]. The results are shown in Table III.

TABLE III. Result of GNB, LR, KNN and nuSVC challenge

	GNB				LR				KNN				nuSVC			
Seq	TN	TP	FN	FP	TN	TP	FN	FP	TN	TP	FN	FP	TN	TP	FN	FP
1	10249	5262	342	0	9955	5262	636	0	10136	5262	455	0	10023	5262	568	0
2	9157	6330	366	0	8825	6330	698	0	9019	6330	504	0	8894	6330	629	0
3	8388	7152	313	0	8026	7152	675	0	8246	7152	455	0	8108	7152	593	0
4	7691	7863	265	34	7338	7897	618	0	7574	7897	382	0	7429	7897	527	0
5	10249	5248	356	0	9955	5248	650	0	10136	5248	469	0	10023	5248	582	0
6	9157	6254	442	0	8825	6254	774	0	9019	6254	580	0	8894	6254	705	0
7	8388	6979	486	0	8026	6979	848	0	8246	6979	628	0	8108	6979	766	0
8	7725	7638	490	0	7338	7638	877	0	7574	7638	641	0	7429	7638	786	0
9	10249	5241	363	0	9955	5241	657	0	10136	5241	476	0	10023	5241	589	0
10	9157	6223	473	0	8825	6223	805	0	9019	6223	611	0	8894	6223	736	0
11	8388	6920	545	0	8026	6920	907	0	8246	6920	687	0	8108	6920	825	0
12	7725	7536	592	0	7338	7536	979	0	7574	7536	743	0	7429	7536	888	0
13	10249	5241	363	0	9955	5241	657	0	10136	5241	476	0	10023	5241	589	0

Seq	GNB				LR				KNN				nuSVC			
	TN	TP	FN	FP	TN	TP	FN	FP	TN	TP	FN	FP	TN	TP	FN	FP
14	9157	6222	474	0	8825	6222	806	0	9019	6222	612	0	8894	6222	737	0
15	8388	6919	546	0	8026	6919	908	0	8246	6919	688	0	8108	6919	826	0
16	7725	7532	596	0	7338	7532	983	0	7574	7532	747	0	7429	7532	892	0

Formula (9) and (10) were used to determine the accuracy and F1-score with data source from Table III. The calculation results were visualized with Fig. 6.a for the accuracy and Fig. 6.b. for the F-1 score.

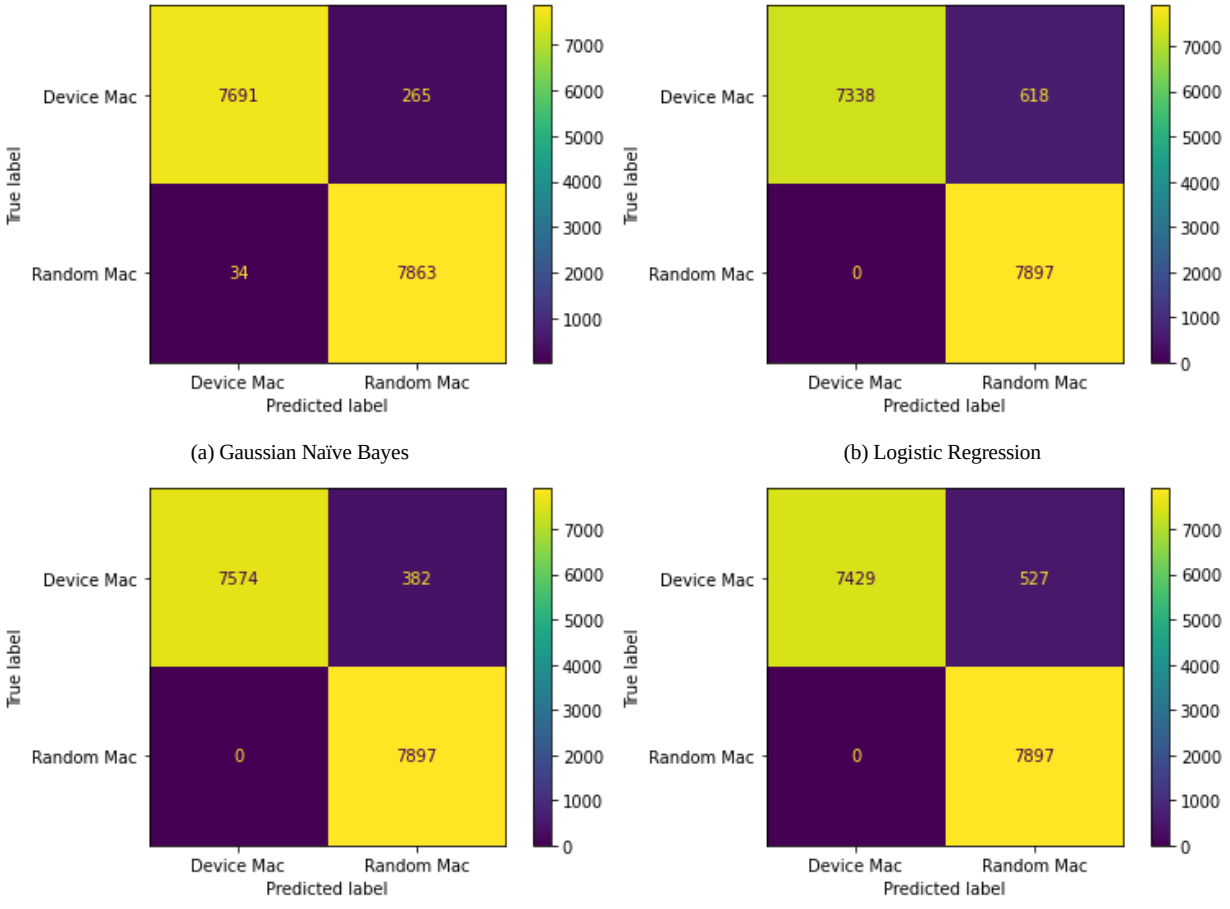
(a) accuracy (b) F1-Score

Figure 6. Accuracy and F1-Score

The highest accuracy 98,1139% and the F1 score 98,1342% was on the last research sequence of the first group (research number 4) using Gaussian Naïve Bayes (GNB). According to Table I: research sequence numbers 1 to 4 belong to P=50% and the last sequence was M=6. Even though other methods did not have as high an accuracy and F1 score as GNB, they all have the same peak point in the fourth research.

The numbers of TP, FN, FP, and TN in research sequence number 4 shown in Table III are represented by the confusion matrix in Fig. 7. True means the sample was correctly identified by the classifier, and false means the positive sample was incorrectly identified.

TP means that the predicted label of a random MAC address is the same as the true label since the classification is a random MAC address or else. TP is at the bottom right of the confusion matrix, while TN is at the top left. The value of TP in the GNB is 7,863 data rows.



(a) Gaussian Naïve Bayes

(b) Logistic Regression

(c) K-nearest neighbors

(d) New Support Vector Classification

Figure 7. Confusion matrix of the highest accuracy

If P and M values came from the highest accuracy substituted in (1), then the random MAC address classification rule can be expressed in (11):

$$y_i = \begin{cases} 1, if \ \left(\frac{\sum_j loginnum_j<6}{|MAC\_Address|}\right)_i \geq 50\% \ and \ \mu login_i < 6; \\ 0, \qquad otherwise. \end{cases} \quad (11)$$

Formula (11) can be expressed as algebra (12):

$$\pi_{SUM(loginnum<6)/COUNT(1)>=0.5 \ AND \ AVG\ (loginnum)<6 \rightarrow israndomclass} \gamma_{SUM\ (loginnum<6),COUNT\ (1),AVG\ (loginnum)} \sigma_{oui = DEVICE\_OUI\_}\ (radrandomisemacfactor) \quad (12)$$

7. Conclusion

The results of research using the Gaussian Naïve Bayes, Logistic Regression, K-Nearest Neighbor, and New Support Vector Classification show that the best accuracy of 98.1139% and the best F1-score of 98.1342% was obtained from P=50% and M=6. It is possible to do quick random MAC address detection based on OUI on a captive portal in the real world using (12). Further research is expected to improve the accuracy with many scenarios, such as adjusting the parameter, adjusting the classification rule, and introducing another classifier.

References

[1] "Individuals using the Internet - ITU Datahub," Individuals using the Internet - ITU Datahub. Accessed: Nov. 23, 2023.

E-mail:author's email

[Online]. Available: https://datahub.itu.int/data/?e=701&c=&i=11624

[2] E. M. Caudill and P. E. Murphy, "Consumer Online Privacy: Legal and Ethical Issues," *Journal of Public Policy & Marketing*, vol. 19, no. 1, pp. 7–19, Apr. 2000, doi: 10.1509/jppm.19.1.7.16951.

[3] airheadsteam@hpe.com, "Mac Address Randomization How To Tackle It With Aruba Infrastructure." 2020. [Online]. Available: https://www.arubanetworks.com/assets/tg/TD_Mac-Address-Randomization.pdf

[4] "IEEE Recommended Practice for Privacy Considerations for IEEE 802(R) Technologies," IEEE. doi: 10.1109/IEEESTD.2020.9257130.

[5] J.-C. Zúñiga, C. J. Bernardos, and A. Andersdotter, "Randomized and Changing MAC Address." 2023. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-madinas-mac-address-randomization-06

[6] J. Henry and Y. Lee, "Randomized and Changing MAC Address Use Cases and Requirements." 2023. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-madinas-use-cases/05/

[7] T. Aytac, M. A. Aydin, and A. H. Zaim, "Detection DDOS Attacks Using Machine Learning Methods," *Electrica*, vol. 20, no. 2, pp. 159–167, Jun. 2020, doi: 10.5152/electrica.2020.20049.

[8] R. Pandey, M. Pandey, and A. Nazarov, "Enhanced DDoS Detection using Machine Learning," in *2023 6th International Conference on Information Systems and Computer Networks (ISCON)*, Mathura, India: IEEE, Mar. 2023, pp. 1–4. doi: 10.1109/ISCON57294.2023.10112033.

[9] A. Fadlil, I. Riadi, and S. Aji, "DDoS Attacks Classification using Numeric Attribute-based Gaussian Naive Bayes," *ijacsa*, vol. 8, no. 8, 2017, doi: 10.14569/IJACSA.2017.080806.

[10] Z. K. Mrisho, J. D. Ndibwile, and A. E. Sam, "Low Time Complexity Model for Email Spam Detection using Logistic Regression," *IJACSA*, vol. 12, no. 12, 2021, doi: 10.14569/IJACSA.2021.0121215.

[11] S. Akiyama, R. Morimoto, and Y. Taniguchi, "A Study on Device Identification from BLE Advertising Packets with Randomized MAC Addresses," in *2021 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, Gangwon, Korea, Republic of: IEEE, Nov. 2021, pp. 1–4. doi: 10.1109/ICCE-Asia53811.2021.9641870.

[12] I. Zhaika and D. Hay, "Device Identification in the Presence of MAC Randomization," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, Rio de Janeiro, Brazil: IEEE, Dec. 2022, pp. 2086–2091. doi: 10.1109/GLOBECOM48099.2022.10001085.

[13] E. Fenske, D. Brown, J. Martin, T. Mayberry, P. Ryan, and E. Rye, "Three Years Later: A Study of MAC Address Randomization In Mobile Devices And When It Succeeds," *Proceedings on Privacy Enhancing Technologies*, vol. 2021, no. 3, pp. 164–181, Jul. 2021, doi: 10.2478/popets-2021-0042.

[14] L. Casabella, M. D'Anna, and P. A. García-Sánchez, "Apéry Sets and the Ideal Class Monoid of a Numerical Semigroup," *Mediterranean Journal of Mathematics*, vol. 21, no. 1, p. 7, Nov. 2023, doi: 10.1007/s00009-023-02550-8.

[15] A. DeKok and J. Korhonen, "Dynamic Authorization Proxying in the Remote Authentication Dial-In User Service (RADIUS) Protocol," RFC Editor, RFC8559, Apr. 2019. doi: 10.17487/RFC8559.

[16] S. Sutriman and B. Sugiantoro, "Analysis of Password and Salt Combination Scheme To Improve Hash Algorithm Security," *IJACSA*, vol. 10, no. 11, 2019, doi: 10.14569/IJACSA.2019.0101158.

Imam Riadi Prof. Dr. Ir. Imam Riadi, M.Kom. earned his Doctorate degree from Universitas Gadjah Mada in 2014, a Master's degree in Computer Science from Universitas Gadjah Mada in 2004, and a Bachelor's degree in Electrical Engineering Education from Universitas Negeri Yogyakarta in 2001. He is currently serves as a Professor in the field of Information System at the Universitas Ahmad Dahlan, Yogyakarta. His research specializes in Information Security, Digital Forensics, and Network & Cloud Forensics

Abdul Fadlil Abdul Fadlil Prof. Drs. Ir. Abdul Fadlil, M.T., Ph.D. earned his Doctorate degree from Universiti Teknologi Malaysia in 2006, a Master's degree in Electrical Engineering from Universitas Gadjah Mada in 2000, and a Bachelor's degree in Physics from Universitas Gadjah Mada in 1992. He is currently serves as a Professor in the field of Electrical Enginering at the Universitas Ahmad Dahlan, Yogyakarta. His research specializes in Electronics & Instrumentation, Pattern Recognition, and Soft Computing.

Basit Adhi Prabowo Basit Adhi Prabowo, S.T. currently pursuing his Master of Informatics at the Universitas Ahmad Dahlan, with a research focus on Machine Learning and Network Security. He currently serves as head of the IT Department at Universitas Aisyiyah, Yogyakarta.