MAC Address Classification in Privacy Issue Using Gaussian Naïve Bayes

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*

*corr_author: imam.riadi@is.uad.ac.id

Abstract - There have been several initiatives within standards committees to overcome privacy issues, including user tracking activity based on Media Access Control (MAC) addresses. The implementation of randomized MAC addresses on captive portals, with user-specific connection limits to address privacy concerns, introduces some problems. To address this issue, device removal based on OUI classification was proposed. Connection data taken from the RADIUS server were divided into two distinct classes, either random or not. Gaussian Naïve Bayes was utilized to classify the data with 16 distinct thresholds, and the solution with the highest accuracy was selected. The research produced results showing that all classifications had an accuracy above 96%. Values of 6 and 50% for Mac address thresholds and random percentage thresholds gave the highest accuracy of 98.1139%. This indicates that random Mac address classification in the real world can be done using the result.

Keywords: captive portal, Gaussian Naïve Bayes, MAC, privacy, randomization

I. INTRODUCTION

Technology and the internet are necessities for today's modern life. Numerous features of gadgets, such as location, sensors, and applications, can make life easier for humans. A human activity recognition (HAR) system is used in several applications to capture users' physical activity through sensors [1]. Users frequently post the outcomes of their physical activity recordings on social media.

During the Covid-19 pandemic, internet use—especially on social media—grew significantly in the world. Due to their restricted mobility, people used their devices more frequently. Most people used the internet and internet-based services to communicate, engage, and carry out their work from home during the lockdown. Compared to pre-lockdown levels, internet service utilization increased from 40% to 100% [2].

The goal of artificial intelligence development is to assist people in resolving problems. Business is being disrupted by the development of artificial intelligence technology. Incorporated into websites and social media, artificial intelligence has found widespread application in the marketing sector. Products that correspond with the user's online and social media usage habits are displayed by tracking user activity [3].

In the end, user privacy will conflict with these advancements and even with human existence in some cases [4]. Companies that develop browser applications and are concerned about user privacy are including measures that safeguard the privacy. Operating system vendors also put in place safeguards to preserve user privacy. In 2014, random media access control (MAC) addresses were introduced as one of these privacy protection techniques. MAC address randomization was originally implemented by Apple in iOS 8 in 2014. Google added a feature to Android 6.0 a year later and Microsoft corporation made it available on the Windows 10 operating system in 2016 [5].

A recommended practice guideline for integrating privacy into IEEE 802 technology was released by IEEE in 2020 [6]. Additionally, IEEE developed ideas for a range of use applications involving Mac address randomization. The troublesome circumstance of employing random MAC addresses in Captive Portal is brought up in one of the use cases. If a device uses a random address, the Captive Portal will have trouble associating with it. If limits are placed on how many devices can connect to the Captive Portal—for instance, allowing a single user to access via two devices—new issues will surface.

One solution to the issue of device limits is to remove the device from the Captive Portal. The address thought to be a random Mac address is the first to be removed, followed by the address thought to be the device's MAC address. As a result, a technique is required to ascertain whether the MAC address being used randomly.

Research has been conducted in the past to perform classification using Gaussian Naive Bayes (GNB) as follows:

- The brute force technique is one of the methods used in the hacking process. By tracking this brute force method early on, it can be stopped or avoided. Research [7] implemented Naïve Bayes to identify brute force attacks or regular traffic in the network on the Hydra process using the Telnet service using the Wireshark tools
- Support Vector Machine (SVM), Artificial Neural Networks (ANN), Logistic Regression, Gaussian Naive Bayes (GNB), Multinomial Naive Bayes, Bernoulli Naive Bayes, Decision Tree (entropy-Gini), K-nearest Neighbor (KNN), and Random Forest algorithms were used to detect DDoS attacks by analyzing the success rates of threat determination[8]
- Research [9] compared of k-NN and Naïve Bayes Classifier to determine fish freshness
- Employing the Gaussian Naive Bayes method for statistical analysis, research [10] aimed to create a novel method for detecting DDoS attacks based on network traffic activity. Based on the average and standard deviation of network packets according to the Gaussian method, the novel method of detecting DDoS assaults was anticipated to be related to the Intrusion Detection System (IDS) in order to predict the occurrence of DDoS attacks.
- Using Naïve Bayes and the K-Nearest Neighbor approach, the researcher [11] conducted a security evaluation to identify attacks on a website, namely SQL Injection attacks. The testing website exhibits SQL Injection vulnerabilities, according to the results.

A number of research on random MAC addresses was done as follows:

- Research [12] on random MAC addresses on Bluetooth was conducted in 2021. This research applied the time a device disconnects and the time the device reconnects to the network to associate MAC addresses with connected devices
- An innovative passive sniffing de-randomization technique was suggested by the research [13]. A method based on filtering the frames according to the received signal power has been devised for counting the number of persons inside a mass transit vehicle. An accuracy of 75% on average has been achieved in this kind of situation

Based on a threshold value determined by Gaussian Naïve Bayes, this research determines whether the MAC address is random; nevertheless, it did not identify the correlation between random MAC addresses and devices.

II. METHOD

A. Data Acquisition

Research data were obtained from users who logged in via the University's captive portal[14], including lecturers, students, education staffs, and others. Everyone can log in using the same credentials as the account to log in to the Information System. People who do not have an information system account will be given a special card containing account information and password so they can enjoy internet services on campus.

Initially, no users can use internet services, except for devices that are included in the list of devices that can connect to the internet without logging in. A device from a user connected using Wi-Fi will be received by the access point and forwarded to the router. The router checks whether the device is allowed to connect to the internet or not.

The router returns a login page to a user who is not allowed to connect to the internet. The user enters credentials, then they are sent back to the hotspot. The router communicates to the RADIUS server to validate the received credential. The router sends login information to the RADIUS server and allows packets from the user to pass through to the internet.

There were 29 pieces of metadata of the device connected to the captive portal. The data were processed in this research to obtain a summary of the required information based on the user and the MAC address of the device used by the user, with an aggregate count of user logins as exemplified in Table I.

The formula to determine that the Mac address used by the user is a random Mac address (true) or device MAC address (false) is shown in (1).

$$israndom_j = |login| < M \quad (1)$$

where j is iterator of user data; $israndom_j$ is random mac address or not per user per mac address; |login| is count of user login per user per mac address, | . | denotes cardinality[15]; and M is device mac address threshold

TABLE I
EXAMPLE OF RECAPITULATION OF USER DATA

No	User	Mac Address	Login Count	Is Random By User
1	Mr A	12:34:56:12:34:56	15	true/false
2	Mr O	12:34:56:12:34:56	1	true/false
3	Mr O	AB:CD:EF:11:44:66	1	true/false
4	Mrs C	A1:23:CD:56:34:12	9	true/false

Random MAC address classification can be determined using a formula as shown in (2) and (3).

$$y_i = \begin{cases} 1, if\ \%RandomMAC_i \geq P\ \ and\ \ \mu\, login_i < M\ ; \\ 0, \qquad otherwise \end{cases} \tag{2}$$

$$\%RandomMAC_i = \frac{f}{n}.100\% = \left(\frac{|israndom|}{|macaddress|}\right)_i.100\% \tag{3}$$

where i is iterator of OUI; y_i is mac address classification per OUI; %RandomMAC$_i$ is percentage of random mac address per OUI in 0 to 1 scale; P is random mac address percentage threshold; μ login$_i$ is average of login counts per OUI; M is device mac address threshold; f is frequency; n is all members; |israndom| is count of random mac address; and |macaddress| is count of total mac address with identical OUI

The data in Table I were then grouped based on OUI—the first 24 bits or 6 hexadecimal digits of a MAC address—with aggregates: random MAC address counts, member counts, and an average of login counts using SQL query as shown in Fig. 1 based on (2) and (3). Performing query as on Fig. 1 can produce results shown in Table II.

The data in Table II was taken to become training data and saved into the file_datasettraining file with criteria

a. OUI is used by more than one user
b. Random MAC address classification has a maximum login count of five times
c. MAC address classification of the device has a minimum login count of eight times

Data that did not match the criteria was processed with P values of 50%, 68%, 90%, and 95% and M values of 3, 4, 5, and 6. These data are saved respectively into 16 files of file_datasettest for research in the sequence as in Table III.

```
SELECT get_oui(callingstationid) AS oui,
SUM(loginnum < M)/COUNT(1) AS pctRand,
AVG(loginnum) AS LoginNumAvg,
SUM((loginnum < M)/COUNT(1)) >= P AND AVG(loginnum) < M AS isRandomClass
FROM radrandomisemacfactor
WHERE get_voui(callingstationid) NOT IN (SELECT voui FROM `research.validation`)
GROUP BY get_voui(callingstationid);
```

Fig. 1 SQL query

TABLE II
EXAMPLE OF CLASSIFICATION BASED ON OUI

No	OUI	MAC Address Count	Israndom Count (|israndom|)	%RandomMAC (0 to 1 scale)	Login AVG (μ_{login})	Class (y)
1	12:34:56	2	1	$\frac{1}{2}$	8	rand/device
2	AB:CD:EF	1	1	$\frac{1}{1}$	1	rand/device
3	A1:23:CD	1	0	$\frac{0}{1}$	1	rand/device

TABLE III
EXPERIMENT SEQUENCE NUMBER

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

B. Classification Process

The next step uses Gaussian Naïve Bayes to test the classification of 16 test files of file datasettest. Bayes' theorem is expressed through a mathematical as in (4)

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \tag{4}$$

where A is event of A; B is event of B; P(A) is the probability of event A being independent of event B; P(B) is the likelihood that event B will occur without reference to event A; P(A|B) is probability of event A

if B occurs; and P(B|A) is probability of event B in the case that A happens

Gauss introduced the Gaussian distribution in his study of error theory, and it is one of the most used and significant techniques in probability calculation and statistics[10]. The Gaussian distribution can be obtained through the equation as shown in (5).

$$f(x) = P(x_i|y) = \frac{1}{\delta\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i-\mu}{\delta}\right)^2} \quad (5)$$

where μ is average; e is Euler constant (2.71828); π is ratio of the circumference of a circle to its diameter (3.14159); and δ is standard deviation

The mean (μ) and standard deviation (δ) were obtained through (6) and (7).

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n} \quad (6)$$

$$\delta = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\mu)}{n}} \quad (7)$$

where n is all of the group members

How well the classification model works can be seen from the accuracy and F1-score value with a range from 0 (low) to 1 (high). Accuracy is the number of correct predictions, both true positives and true negatives, for all data. Accuracy is used to evaluate overall performance using (8). Meanwhile, the F1 score is used to measure the harmonic mean of precision and sensitivity/recall as in (9) [16].

$$A = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

$$F1 = \frac{TP}{TP+\frac{(FP+FN)}{2}} \quad (9)$$

where TP is true positive; TN is true negative; FP is false positive; FN is false negative;

Process to obtain the M and P values for each file_datasettest file was carried out with the following steps:

a. Read each dataset:
   - $dataset_{test}$:= read_file(file_datasettest)
   - $dataset_{training}$:= read_file(file_datasettraining)
b. Balance $dataset_{test}$ for each class
c. Normalize the data so that the average login count (loginavg) on a scale of 0 to 1
   - dataset:= concat($dataset_{test\_parameter}$[“loginavg”], $dataset_{training\_parameter}$[“loginavg”])
   - dataset:= min_max_scaler(dataset)
   - $dataset_{test\_parameter}$[“loginavg”], $dataset_{training\_parameter}$[“loginavg”] := split(dataset)
d. Split each dataset into parameter and classification data as shown in Fig. 2:
   - $dataset_{test\_parameter}$, $dataset_{test\_class}$:= split($dataset_{test}$)
   - $dataset_{training\_parameter}$, $dataset_{training\_class}$:= split($dataset_{training}$)
e. Run classification predictions with Gaussian Naïve Bayes
   - model.fit($dataset_{training\_parameter}$, $dataset_{training\_class}$, “Gaussian Naïve Bayes”)
   - $dataset_{predict\_class}$:= model.predict($dataset_{test\_parameter}$)
   - accuracy := accuracy_score($dataset_{test\_class}$, $dataset_{predict\_class}$)
   - f1 := f1_score($dataset_{test\_class}$, $dataset_{predict\_class}$)

oui	membercount (|member|)	israndomcount (|israndom|)	%RandomMAC (0 to 1 scale)	loginavg (μ_{login})	classification (y)
12:34:56	2	1	$\frac{1}{2}$	0.8	rand/device
AB:CD:EF	1	1	$\frac{1}{1}$	0.1	rand/device
A1:23:CD	1	0	$\frac{1}{0}$	0.1	rand/device

%RandomMAC (0 to 1 scale)	loginavg (μ_{login})
$\frac{1}{2}$	8
$\frac{1}{1}$	1
$\frac{1}{0}$	1

parameter

classification (y)
rand/device
rand/device
rand/device

class

Fig. 2 Dataset splitting

Measure the quality of selected classifications' correlation with the Matthews correlation coefficient (MCC) using (10) [16].

$$MCC = \frac{TPxTN - FPxFN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (10)$$

MCC ranges from -1 to 1. The classifier has a good correlation if the MCC value is near to 1 and the value of 0 means that the classifier is a random guess classifier.

III. RESULT AND DISCUSSION

To classify MAC Address, either random or not, Mac address data were needed to split into two types of datasets: a training dataset and several testing datasets. A training dataset was selected using a rule, and the remaining data became the testing datasets. The training dataset needs to be balanced to get more precise results since GNB is a probabilistic classifier, then the datasets need to be normalized within the range between 0 (probability of 0%) and 1 (probability of 100%).

The training datasets of both random and device MAC addresses were used to find the mean and variance of the features in the training phase. In the testing phase, GNB computes the probability that the MAC address belongs to each class by applying the Gaussian probability density function to each feature of the testing datasets using the mean and variance calculated in the training phase. The likelihood of each Mac address in the testing datasets belonging to the random class is determined by the probability greater than 50%.

A. Data

The research used 26,138 recapitulation data of user data recorded by the radius server in accordance with Table I. This data were then classified based on OUI to become 16,464 data in accordance with Table II. The classification data based on OUI took 611 data as training data with details of 107 device Mac address data and 504 random Mac address data matching the criteria.

The training data was balanced so that the total was 107 device Mac addresses and 107 random Mac addresses saved as a CSV file with filename **research.devmac-randmac.validation.dat.proc.csv**. Classification based on OUI which did not include training data was 15,853. The data was then processed with M and P values according to Table III to obtain 16 test data saved in the CSV files with the filename format: **research.dat.proc.P.M.csv**. The number of random MAC Address and device Mac Address in training and testing data shown in Table IV.

B. Finding M and P

The source code line 24 in Fig. 3 refers to lines 3 to 6 were used to obtain feature data (X_param) and labels (y_cls) on the training data file and each test data file from 16 test data files using the variables percent (P) in line 22 and devThres (M) in line 23. Then GNB processing was triggered by line 27. The training data features and labels were trained using the fit() method in line 13. After the model was trained, predict() method line 14 determines the label prediction for each sample in the test data feature using the variance and mean values of the features obtained in the training process.

TABLE IV
NUMBER OF TRAINING DATA ANF TESTING DATA

Data	Random MAC Address	Device MAC Address
Training Data	107	107
Testing Data Experiment 1	10591	5262
Testing Data Experiment 2	9523	6330
Testing Data Experiment 3	8701	7152
Testing Data Experiment 4	7956	7897
Testing Data Experiment 5	10605	5248
Testing Data Experiment 6	9599	6254
Testing Data Experiment 7	8874	6979
Testing Data Experiment 8	8215	7638
Testing Data Experiment 9	10612	5241
Testing Data Experiment 10	9630	6223
Testing Data Experiment 11	8933	6920
Testing Data Experiment 12	8317	7536
Testing Data Experiment 13	10612	5241
Testing Data Experiment 14	9631	6222
Testing Data Experiment 15	8934	6919
Testing Data Experiment 16	8321	7532

```
1  import pandas as pd
2
3  def readData(percent_threshold_, device_macaddr_threshold_):
4      dtrain = pd.read_csv(r'research.devmac-randmac.validation.dat.proc.csv')
5      dtest  = pd.read_csv(r'research.dat.proc.' + str(percent_threshold_) + '_' + str(device_macaddr_threshold_) + '.csv')
6      return dtrain.drop('isRandomClass', axis=1), dtrain.data['isRandomClass'], dtest.drop('isRandomClass', axis=1),
       dtest.data['isRandomClass']
7
8  def runGaussianNB(X_param_train_, X_param_test_, y_cls_train_, y_cls_test_, device_macaddr_threshold_, conf_matrix_label_):
9      from sklearn.naive_bayes import GaussianNB
10     from sklearn.metrics import ( accuracy_score, confusion_matrix, ConfusionMatrixDisplay, f1_score )
11
12     model = GaussianNB()
13     model_.fit(X_param_train_, y_cls_train_.values.ravel())
14     y_pred = model_.predict(X_param_test_)
15
16     cm     = confusion_matrix(y_cls_test_, y_pred)
17     disp   = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=conf_matrix_label_)
18     disp.plot()
19     accu_y = accuracy_score(y_cls_test_, y_pred)
20     f1     = f1_score(y_cls_test_, y_pred, average="binary")
21
22 percent  = 50
23 devThres = 6
24 X_param_train, y_cls_train, X_param_test, y_cls_test  = readData(percent, devThres)
25
26 label=["Device Mac", "Random Mac"]
27 runGaussianNB(X_param_train, X_param_test, y_cls_train, y_cls_test, devThres, label)
```

Fig. 3 Source code to find M and P (Python)

After each test data of the 16-test data run with Gaussian Naïve Bayes classification with the same training data and adjustment of percent and devThres variable, the confusion matrix was calculated by line 16. The result was the number of classifications with true positive (TP), classification with false positive (FP), classification with true negative (TN), and classification with false negative (FN) can be shown in Table V.

The data in Table IV was used to calculate the accuracy and F1 score in lines 19 and 20. The results of the accuracy calculation are shown in Fig. 4, and the F1 score is shown in Fig. 5.

TABLE V
GNB CLASSIFICATION RESULT

Experiment Number	M	P	Device TN	Random TP	False Device FN	False Random FP
1	3	50%	10249	5262	342	0
2	4	50%	9157	6330	366	0
3	5	50%	8388	7152	313	0
4	6	50%	7691	7863	265	34
5	3	68%	10249	5248	356	0
6	4	68%	9157	6254	442	0
7	5	68%	8388	6979	486	0
8	6	68%	7725	7638	490	0
9	3	90%	10249	5241	363	0
10	4	90%	9157	6223	473	0
11	5	90%	8388	6920	545	0
12	6	90%	7725	7536	592	0
13	3	95%	10249	5241	363	0
14	4	95%	9157	6222	474	0
15	5	95%	8388	6919	546	0
16	6	95%	7725	7532	596	0

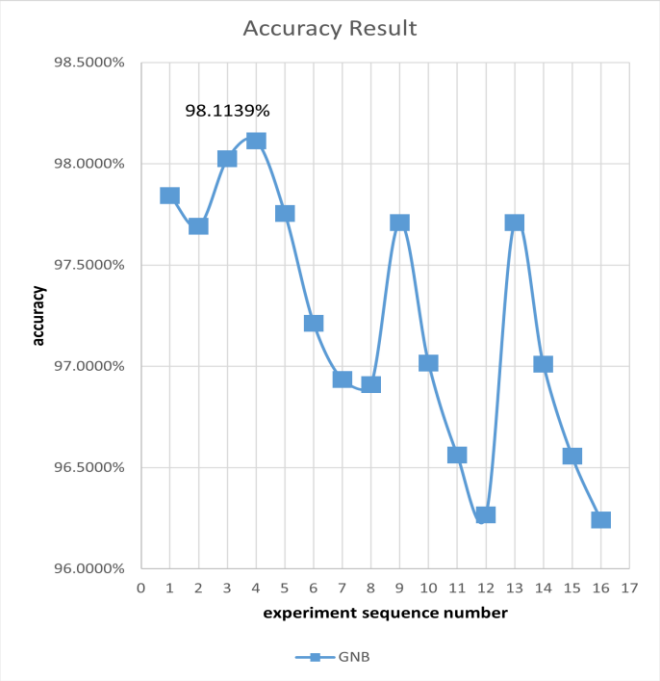

Fig. 4 Accuracy of the classification

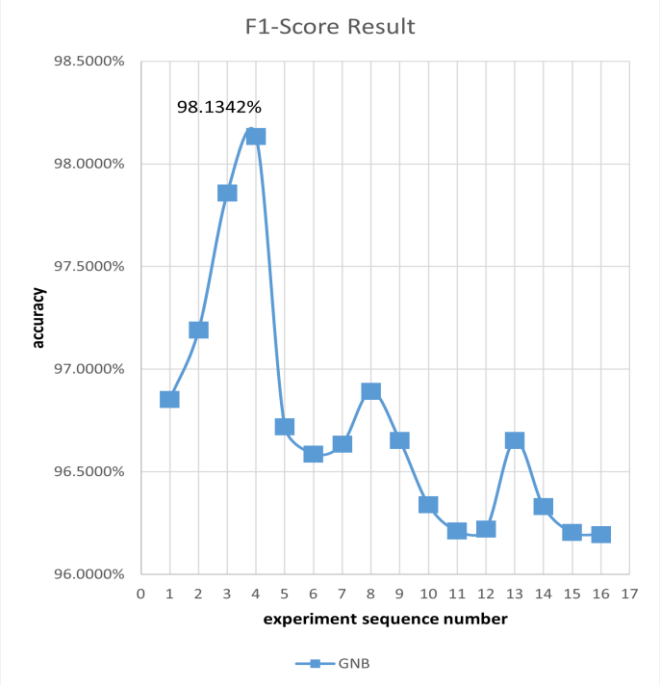

Fig. 5 Classification's F1-score values

The two graphs show that the highest accuracy and F1 value were obtained in sequence number 4. The accuracy of sequence number 4 was 98.1139%, while the F1 value of sequence number 4 was 98.1342%. The random percentage threshold (P) in sequence number 4 was 50%. The device MAC address threshold (M) in sequence number 4 was 6. The confusion matrix from sequence number 4 can be seen in Fig. 6.

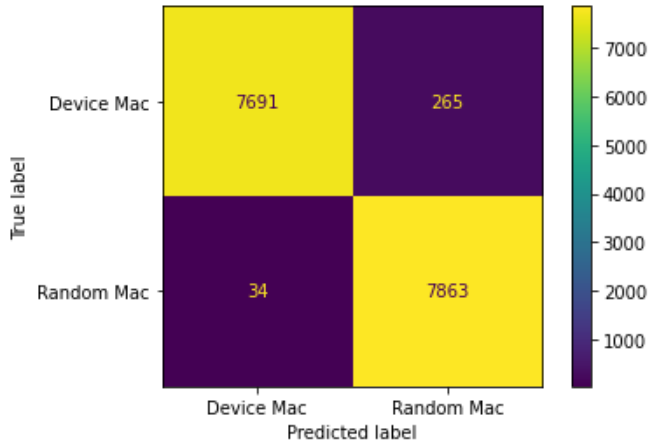

Fig. 6 Confusion matrix sequence number 4

Calculate the classification quality using MCC

$$MCC = \frac{TPxTN - FPxFN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} = \frac{7863x7691 - 34x265}{\sqrt{(7863+34)(7863+265)(7691+34)(7691+265)}}$$

$$MCC = \frac{60474333 - 9010}{\sqrt{7897x8128x7725x7956}} = \frac{60465323}{62808662.86} = 96.26\%$$

MCC result is near to 1, which means that the classification correlation is good.

IV. CONCLUSION

The results of research using the Gaussian Naïve Bayes method show that a random percentage threshold (P) of 50% and a device MAC address threshold (M) of 6 can be used to help decide whether the OUI group is used for random MAC addresses or not with the accuracy of 98.1139%, F1-score of 98.1342%, and MCC of 96.26%. This decision can be used for further processes, including eliminating the MAC addresses of devices that are still active on the captive portal if the user has exceeded the permitted device usage limit. Further researchers are expected to proceed with more than two parameters, using different formulas or using other classifiers.

REFERENCES

[1] M. Straczkiewicz, P. James, and J.-P. Onnela, "A systematic review of smartphone-based human activity recognition methods for health research," *NPJ Digit Med*, vol. 4, no. 1, p. 148, Oct. 2021, doi: 10.1038/s41746-021-00514-4.

[2] R. De', N. Pandey, and A. Pal, "Impact of digital surge during Covid-19 pandemic: A viewpoint on research and practice," *Int J Inf Manage*, vol. 55, p. 102171, Dec. 2020, doi: 10.1016/j.ijinfomgt.2020.102171.

[3] S. Chintalapati and S. K. Pandey, "Artificial intelligence in marketing: A systematic literature review," *International Journal of Market Research*, vol. 64, no. 1, pp. 38–68, Jan. 2022, doi: 10.1177/14707853211018428.

[4] K. Salmani and B. Atuh, "A Lesson for the Future: Will You Let Me Violate Your Privacy to Save Your Life?," *Journal of Cybersecurity and Privacy*, vol. 3, no. 2, pp. 259–274, Jun. 2023, doi: 10.3390/jcp3020014.

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

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

[7] T. Amijoyo, R. Umar, and A. Yudhana, "Bruteforce In The Hydra Process And Telnet Service Using The Naïve Bayes Method," *Jurnal Mantik*, vol. 4(1), pp. 319–326, 2020.

[8] 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.

[9] A. Yudhana, R. Umar, and S. Saputra, "Fish Freshness Identification Using Machine Learning: Performance Comparison of k-NN and Naïve Bayes Classifier," *Journal of Computing Science and Engineering*, vol. 16, no. 3, pp. 153–164, Sep. 2022, doi: 10.5626/JCSE.2022.16.3.153.

[10] A. Fadlil, I. Riadi, and S. Aji, "Review of Detection DDOS Attack Detection Using Naive Bayes Classifier for Network Forensics," *Bulletin of Electrical Engineering and Informatics*, vol. 6, no. 2, pp. 140–148, Jun. 2017, doi: 10.11591/eei.v6i2.605.

[11] H. Herman, I. Riadi, and Y. Kurniawan, "Vulnerability Detection With K-Nearest Neighbor and Naïve Bayes Method Using Machine Learning," 2023, doi: 10.29099/ijair.v7i1.795.

[12] L. Jouans, A. C. Viana, N. Achir, and A. Fladenmuller, "Associating the Randomized Bluetooth MAC Addresses of a Device," in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA: IEEE, Jan. 2021, pp. 1–6. doi: 10.1109/CCNC49032.2021.9369628.

[13] M. Uras, E. Ferrara, R. Cossu, A. Liotta, and L. Atzori, "MAC address de-randomization for WiFi device counting: Combining temporal- and content-based fingerprints," *Computer Networks*, vol. 218, p. 109393, Dec. 2022, doi: 10.1016/j.comnet.2022.109393.

[14] F. A. Mustika, F. P. Sulistyo, and C. A. Tanof, "Implementasi System Captive Portal Dengan Otentikasi RADIUS," *Jurnal Ilmiah FIFO*, vol. 12, no. 1, p. 49, Jul. 2020, doi: 10.22441/fifo.2020.v12i1.005.

[15] 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, Jan. 2024, doi: 10.1007/s00009-023-02550-8.

[16] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, p. 6, Dec. 2020, doi: 10.1186/s12864-019-6413-7.