REPOSITOR, Vol. 3, No. 4, Agustus 2021, Pp. 367-374
ISSN : 2714-7975
E-ISSN : 2716-1382

Classification Of *Malware* Families Using *Naïve Bayes* Classifier

Ramadan Pratama[*1], Denar Regata Akbi[2], Vinna Rahmayati Setianing Nastiti[3]
[1,2,3]Universitas Muhammadiyah Malang
ramadan_437253@webmail.umm.ac.id[*1], dnarregata@umm.ac.id[2], vinastiti@umm.ac.id[3]

Abstrak

Dikarenakan peningkatan pengguna smartphone Android berbanding lurus dengan peningkatan pengembangan malware yang semakin pesat. Tidak jarang penelitian tentang malware setiap tahunnya yang membahas tentang malware families dengan berbagai macam pendekatan yang salah satunya machine learning. Dengan mendapatkan data malware yang kredibel, dapat memudahkan peneliti dalam menganalisa malware. Terdapat kumpulan data malware yang dibuat the Canadian Institute for Cybersecurity (CIC) yang dapat diakses secara publik. Data ini disebut CICInvestAndMal2019 yang berisi data malware. Dataset ini dibuat dengan melakukan analisa statis dan dinamis pada smartphone secara real time. Hasil dari analisa tersebut kemudian diproses dengan metode Random Forest yang menghasilkan precision 61.2% dan recall 57.7%. Berdasarkan penelitian tersebut, maka penulis akan mengklasifikasikan dataset CICInvestAndMal2019 menggunakan metode Naïve Bayes, dan hasil yang didapat dari klasifikasi Naïve Bayes adalah nilai recall dan precision sebesar 68% dan 66%.

Kata Kunci: *Machine Learning, Klasifikasi Naïve Bayes, Pearson Product-Moment Correlation, Klasifikasi Malware, CICInvestAndMal2019*

Abstract

Due to the increase in Android smartphone users, it is directly proportional to the rapid increase in malware development. It is not uncommon for research on malware every year to discuss malware families with various approaches, one of which is machine learning. By getting credible malware data, it can make it easier for researchers to analyze malware. There is a publicly accessible malware database created by the Canadian Institute for Cybersecurity (CIC). This data is called CICInvestAndMal2019 which contains malware data. This dataset is created by performing static and dynamic analysis on a smartphone in real time. The results of the analysis were then processed using the Random Forest method which resulted in a precision of 61.2% and a recall of 57.7%. Based on this research, the author will classify the CICInvestAndMal2019 dataset using the Naïve Bayes method, and the results obtained from the Naïve Bayes classification are recall and precision values of 68% and 66%, respectively.

Keywords: *Machine Learning, Naïve Bayes Classifier, Pearson Product-Moment Correlation, Malware Classifier, CICInvestAndMal2019*

1. Pendahuluan

Meningkatnya perkembangan ilmu pengetahuan dan teknologi pada beberapa tahun belakangan ini, telepon genggam atau *handphone* telah berkembang menjadi telepon pintar atau lebih dikenal dengan sebutan *smartphone* yang menawarkan fitur yang lebih canggih dibanding sebelumnya[1]. Agar *smartphone* dapat berfungsi dengan baik, *smartphone* harus terinstall Sistem Operasi/*Operating System* (OS) di dalamnya. Saat ini tersedia beberapa OS yang dapat digunakan pada *smartphone* seperti iOS dari *Apple*, *Windows* untuk *Windows phone*, *Blackberry* OS untuk *Blackberry*, dan Android. Saat ini, jumlah pengguna sistem operasi yang paling banyak digunakan adalah android dan iOS dengan total pengguna 96% (50% android dan 46% iOS) [2].

Seiring meningkatnya pengguna *smartphone* Android, mengakibatkan pengembangan *malware* yang juga semakin pesat[3]. Pada umumnya, *malicious software* atau lebih dikenal dengan *malware* (perangkat lunak berbahaya) didefinisikan sebagai program yang dibuat untuk merusak kinerja dari sebuah OS atau bertujuan untuk mencuri data yang ada dalam sebuah perangkat komputer atau *smartphone*. *Malware* dapat berisikan virus, *worms*, *backdoors*, *Trojan horses* dll[4][5]. Dikarenakan peningkatan penggunaan sistem operasi Android, para pembuat *malware* merancang sebagian besar *malware* untuk menyerang OS Android [6]. Salah satu

pendekatan untuk mendeteksi *malware* adalah dengan menggunakan klasifikasi *malware* dengan *machine learning* [7][8].

Salah satu dari metode *machine learning* adalah *Naïve Bayes* merupakan algoritma yang menggunakan penggolongan statistik sederhana berdasarkan teorema bayes dengan asumsi saling bebas atau *conditional independence*[9].

Data yang akan digunakan diperoleh dari *website Canadian Institute for Cybersecurity*. *website* tersebut adalah *website* yang dimiliki oleh *The university of New Brunswick* yang menyediakan dataset *malware* yang diberi nama *CICInvesAndMal2019* yang terdiri dari 305 sampel *malware*, 918 fitur, serta terdapat 39 jenis *malware family*[10].

Berdasarkan latar belakang tersebut, saya sebagai penulis akan melakukan pengklasifikasian *malware* berdasarkan jenis familinya dengan menggunakan data *CICInvesAndMal2019*. Dataset yang digunakan pada penelitian ini sama dengan yang digunakan pada penelitian Achmad Rizal Yogaswara dkk di tahun 2020 dengan artikelnya yang berjudul *Malware Familiy Classification using k-Nearest Neighbor (k-NN)* [8]. Adapun algoritma klasifikasi *machine learning* yang akan penulis gunakan pada penelitian ini adalah algoritma *Naïve Bayes* (NB) karena dari jurnal[10] dan [8], algoritma yang digunakan pada penelitian sebelumnya adalah *Random forest* dan *k-NN*, dan penulis hendak membandingkan performa dari *Naïve Bayes* (NB), untuk mengetahui seberapa handal metode *Naïve Bayes* dalam proses klasifikasi *malware family* pada dataset *CICAndMal2019*, serta membandingkan hasil dari algoritma *Naïve Bayes*, *k-NN*, dan *Random forest*. Mengetahui fitur-fitur yang berpengaruh dalam dataset *CICInvesAndMal2019*.

Sebelum memasuki tahap klasifikasi, dataset akan memasuki tahap *preprocessing* terlebih dahulu. *Preprocessing* adalah sebuah proses pembersihan, reduksi, dan diskritisasi data. Fase *preprocessing* berikut akan membuat dataset lebih presisi dengan menggunakan metode *preprocessing* data seperti penghapusan bising, ekstraksi fitur dan penghapusan atribut menggunakan metode *Pearson product-moment correlation coefficient* [11].

Algoritma *Pearson correlation coefficient* adalah ukuran hubungan linear antar variabel dan dapat memiliki nilai antara -1 dan 1 [12]. Nilai r didapat dengan cara membagi kovarians dari variabel A dan B dengan akar dari hasil kali antara varian variabel A dan B, dapat dilihat Persamaan 1 berikut.

$$r = \frac{n\,(\Sigma XY) - (\Sigma X)(\Sigma Y)}{\sqrt{[n(\Sigma X^2) - (\Sigma X)^2][n(\Sigma Y^2) - (\Sigma Y)^2]}} \quad (1)$$

Dimana:
r = Pearson correlation coefficient
n = jumlah dari skor yang berpasangan
X = skor dari variable pertama
Y = skor dari variable kedua

Pemilihan metode *pearson correlation coefficient* didasari oleh dataset yang digunakan pada penelitian ini adalah data kontinu sehingga metode *pearson correlation coefficient* dianggap cocok untuk melakukan *selection fiture* dengan membandingkan korelasi antar variabel dengan ***r*** sebagai parameter [13]. Dalam dataset *CICInvesAndMal2019* terdapat 918 fitur, yang masing-masing fitur akan dilihat nilai korelasi terhadap fitur yang lain. Fitur yang memiliki korelasi rendah atau mendekati 0 dan fitur yang tidak berkorelasi atau sama dengan 0 akan dihapus, sedangkan fitur yang memiliki nilai korelasi tinggi atau mendekati -1 dan 1 akan digunakan untuk memasuki tahap klasifikasi [12], [14], [15].

Naïve Bayes merupakan algoritma yang menggunakan penggolongan statistik sederhana berdasarkan teorema Bayes dengan asumsi antar fitur bersifat *conditional independence* atau tidak terikat satu sama lain[9]. Secara matematis, teorema Bayes dapat hitung dengan membagi hasil kali dari probabilitas dari data D ketika diberikan data h dan probabilitas dari hipotesis dengan probabilitas dari data. Persamaan naïve bayes dapat dilihat pada Persamaan 2.

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \quad (2)$$

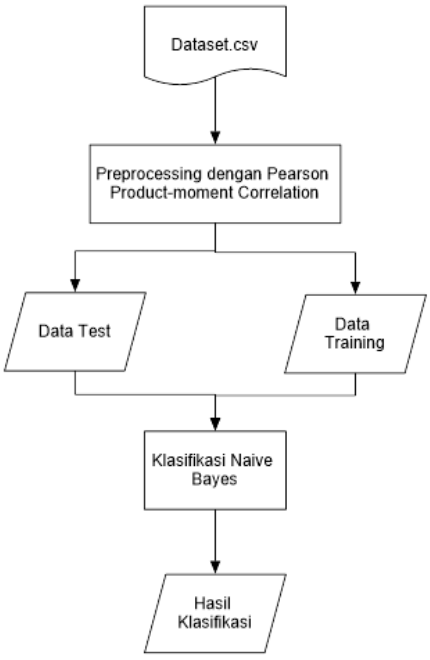
Dimana:
P(h) : probabilitas dari hipotesis. Disebut dengan prior probability dari hipotesis h.
P(D) : probabilitas dari data. Disebut dengan prior probability dari data D.
P(D|h) : probabilitas dari data D ketika diberikan data h. Disebut posterior probability
P(h\D) : probabilitas dari hipotesis h ketika diberikan data D. Disebut posterior probability.

Pemilihan metode *pearson correlation coefficient* didasari oleh dataset yang digunakan pada penelitian ini adalah data kontinu sehingga metode *pearson correlation coefficient* dianggap cocok untuk melakukan selection fitur dengan membandingkan korelasi antar variabel [13]. Dalam dataset *CICInvesAndMal2019* terdapat 918 fitur, yang masing-masing fitur akan dilihat nilai korelasi terhadap fitur yang lain. Fitur yang memiliki korelasi rendah atau mendekati 0 dan fitur yang tidak berkorelasi atau sama dengan 0 akan dihapus, sedangkan fitur yang memiliki nilai korelasi tinggi atau mendekati -1 dan 1 akan digunakan untuk memasuki tahap klasifikasi [12], [14], [15].

2. Metode Penelitian

Pada tahap ini akan menjelaskan analisis dan perancangan sistem penelitian yang akan dilakukan. Bab ini terdiri dari metode, alur, bagan, dan contoh sederhana. Metode dalam penelitian ini terdiri dari pengumpulan data, preprocessing data dengan *Pearson product-moment correlation coefficient*, dan klasifikasi menggunakan *Naïve Bayes*.

Untuk melakukan klasifikasi *malware family* dengan *Naïve Bayes*, maka hal pertama yang dilakukan adalah dengan melakukan studi literatur untuk menentukan algoritma apa yang akan dipakai. Setelah mendapatkan algoritma yang akan digunakan, selanjutnya melakukan pengumpulan data. Jika dataset sudah didapatkan, tahap berikutnya adalah melakukan *preprocessing* data. Setelah melalui tahap preprocessing data, data dibagi menjadi *training* data dan *testing* data. Selanjutnya data tersebut akan masuk ke tahap pengujian dengan metode klasifikasi yang sudah ditentukan sebelumnya. Pada tahap terakhir, jika hasil sudah didapat maka hasil tersebut akan dievaluasi Kembali, seperti pada Gambar 1.



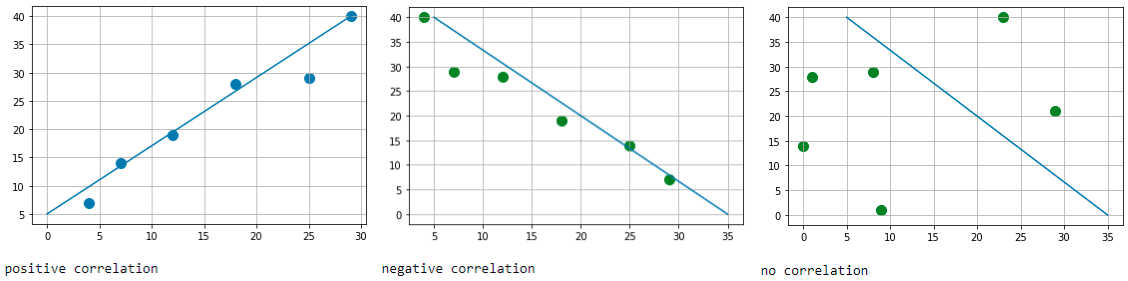
Gambar 1. Skema implementasi dan pengujian

2.1 *Dataset*

Data yang digunakan dalam penelitian ini diunduh dari *website The University of New Brunswick* yang menyediakan dataset *malware* dengan dengan nama *CICinvesAndMal2019*. Dalam dataset tersebut terdapat 305 sampel *malware* dengan masing-masing memiliki 918 atribut.

2.2 *Preprocessing*

Preprocessing dilakukan untuk mengetahui fitur mana yang paling berpengaruh dalam proses klasifikasi agar mendapatkan hasil yang baik saat proses klasifikasi menggunakan *Naïve Bayes*. Hasil Gambar 2 dari *preprocessing* dengan *Pearson product-moment correlation coefficient* menjadikan dataset yang memiliki atribut 918 menjadi 433 atribut. Algoritma *Pearson correlation coefficient* adalah ukuran seberapa besar hubungan linear antar *variable*/(dalam penelitian ini adalah *feature*) ukuran tersebut dapat memiliki nilai antara -1 dan 1 yang menunjukkan kekuatan hubungan antar variabel. Nilai 1 menandakan hubungan sebuah variabel berhubungan kuat positif atau disebut *positive correlation*, sedangkan nilai 0 berarti antar variabel tersebut tidak berhubungan sama sekali disebut *no correlation*. Dan nilai -1 menandakan berhubungan kuat negatif atau disebut *negative correlation*[16].



(a) ***(b)*** ***(c)***

Gambar 2. (a)(b)(c) grafik pearson correlation coefficient

(a) Positive correlation yaitu mendekati atau sama dengan 1
(b) Negative correlation yaitu nilai r mendekati atau sama dengan -1
(c) No correlation yaitu nilai r mendekati atau sama dengan 0

Tahap setelah *preprocessing* data adalah pembagian data menjadi data *training* dan data *testing*. Pada penelitian ini terdapat 4 jenis pembagian data, ditunjukkan pada Tabel 1.

Tabel 1. Perbandingan training data dan test data

Persentase	Training Data	Test Data
60% : 40%	183 data *malware*	122 data *malware*
70% : 30%	213 data *malware*	92 data *malware*
80% : 20%	244 data *malware*	61 data *malware*
90% : 10%	274 data *malware*	31 data *malware*

Dari pembagian data tersebut, akan dilanjutkan dengan klasifikasi menggunakan *Naïve Bayes* dan evaluasi hasil dari klasifikasi menggunakan *confusion matrix* dengan parameter nilai *recall* dan *precision*. *Recall* dihitung dengan menggunakan rumus pada pada gambar dan *Precision* dihitung menggunakan rumus pada Persamaan 3 dan Persamaan 4.

$$\text{Recall} = \frac{TP}{(TP+FN)} \tag{3}$$

$$\text{Precision} = \frac{TP}{(TP+FP)}. \tag{4}$$

3. Hasil Penelitian dan Pembahasan

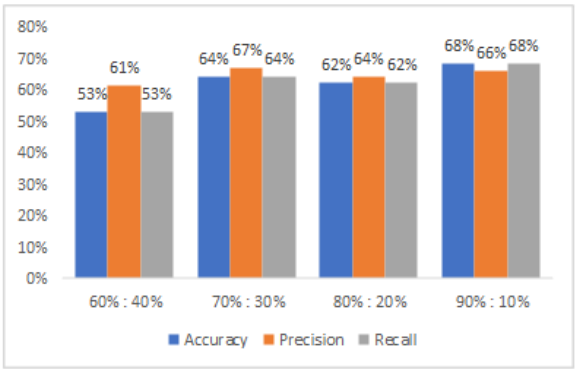
Pada bab ini merupakan tahap implementasi dari rancangan penelitian klasifikasi *malware* family dengan metode *Naïve Bayes*.

3.1 *Pengujian klasifikasi Naïve Bayes dengan komposisi data training dan data testing berbeda*

Data yang sudah melalui *preprocessing* akan memasuki tahap klasifikasi dengan *Naïve Bayes*. Hasil dari klasifikasi dapat dilihat di Tabel 2 dan Gambar 3.

Tabel 2. Hasil klasifikasi

Pembagian data (Data training : data testing)	*Precision*	*Recall*
60% : 40%	61%	53%
70% : 30%	67%	64%
80% : 20%	64%	62%
90% : 10%	66%	68%



Gambar 3. Grafik hasil pengujian naïve bayes

Dari keempat pengujian yang sudah dilakukan, dapat dilihat bahwa pembagian 90% data *training* dan 30% data *testing* mendapatkan nilai *Accuracy* dan *Recall* tertinggi dengan nilai 68% untuk keduanya. Hal ini berarti kedekatan antara nilai yang ingin diprediksi dengan nilai sebenarnya sebesar 68% dan tingkat keberhasilan metode *Naïve Bayes* dalam menemukan Kembali sebuah informasi yaitu sebesar 68%. Sedangkan pembagian 70% data *training* dan 30% data *testing* mendapatkan nilai *precision* tertinggi yaitu 67% yang berarti ketepatan antara data yang diinginkan dan jawaban yang diberikan oleh system sebesar 67%.

Berdasarkan pengujian dengan pembagian data training dan data testing berbeda, terdapat peningkatan pada setiap pembagian data. Hal ini membuktikan bahwa semakin banyak data yang diuji oleh machine learning, maka hasil recall, precision, dan akurasi semakin meningkat.

3.2 *Perbandingan akurasi Naïve Bayes dengan K fold cross validation menggunakan dataset sebelum dan sesudah feature selection*

K-*fold cross validation* atau lebih singkatnya *cross validation* merupakan metode statistik yang digunakan untuk memperkirakan kemampuan sebuah model machine learning dengan cara menggandakan dataset menjadi beberapa kelompok, dan banyaknya kelompok ditentukan dari nilai K. Nilai K dalam penelitian ini bernilai 10, maka kelompok dataset akan berjumlah 10 atau disebut 10 fold cross validation. Selanjutnya dataset dari masing-masing kelompok tersebut akan dibagi 10 dimana 9 untuk data training dan 1 untuk data testing. Tahap berikutnya akan akan dilakukan proses cross validation sebanyak 10 kali dengan penempatan data testing berbeda dari tempat sebelumnya pada kelompok dataset berikutnya [17] [18]. K-fold cross validation dapat dilihat pada Tabel 3 berikut.

Tabel 3. hasil penerapan k-fold cross validation

Pengujian	Dataset tanpa feature selection		Dataset dengan feature selection	
	Fold	Akurasi	Fold	Akurasi
1	Fold - 1	0.68	Fold - 1	0.58
2	Fold - 2	0.48	Fold - 2	0.48
3	Fold – 3	0.55	Fold – 3	0.52
4	Fold – 4	0.55	Fold – 4	0.58
5	Fold – 5	0.58	Fold – 5	0.48
6	Fold – 6	0.57	Fold – 6	0.63
7	Fold – 7	0.47	Fold – 7	0.47
8	Fold – 8	0.67	Fold – 8	0.47

9	Fold – 9	0.60	Fold – 9	0.60
10	Fold – 10	0.63	Fold – 10	0.50
Mean		0.58		0.53

Dari hasil pengujian scenario kedua yang terdapat pada tabel diatas, klasifikasi Naïve Bayes dengan K *fold cross validation* pada dataset yang belum melalui tahap *feature selection* menunjukkan hasil yang lebih baik dengan nilai rata-rata akurasi adalah 58%, dan hasil tertinggi pada fold ke 1 yaitu 68%. Sementara fold tertinggi pada dataset dengan *feature selection* terdapat pada fold ke 6 dengan nilai 63%. Hal ini membuktikan bahwa penerapan *feature selection* tidak berkerja dengan baik dalam peningkatan akurasi terhadap data CICInvesAndMal2019. Karena sekecil apapun pengaruh sebuah feature, tetap akan mempengaruhi keakuratan (*accuracy*) machine learning. Hal ini juga membuktikan bahwa akurasi *pearson correlation coefficient* masih terbilang kurang efektif dalam penentuan feature yang berpengaruh.

3.3 *Perbandingan hasil klasifikasi dengan penelitian sebelumnya*

Dalam subab ini akan dilihat perbedaan hasil Naïve Bayes dengan hasil penelitian sebelumnya yang menggunakan metode random forest dan k-NN dalam pengklasifikasian malware family dengan data *CICInvesAndMal2019*. Hasil dari ketiga metode tersebut dapat dilihat pada tabel dengan menggunakan nilai *recall* dan *precision*.

Tabel 4. Perbandingan precision dan recal pada penelitian sebelumya

Persentase	*Random forest*	*k-NN*	*Naïve Bayes*
Precision :	61%	22%	50%
Recall :	58.7%	22%	49%

Berdasarkan Tabel 4 bahwa hasil klasifikasi random forest lebih baik dari pada Naïve Bayes. Namun metode Naïve Bayes lebih baik dari k-NN. Hal ini dikarenakan data malware kemungkinan tidak cocok untuk diklasifikasikan dengan metode Naïve Bayes. Hal ini dikuatkan dengan scenario 2 dimana hasil akurasi dari klasifikasi sebelum feature selection lebih tinggi dibandingkan setelah feature selection. Serta terjadinya imbalance data karena terdapat kelas yang hanya memiliki 3 data yang apabila dalam pembagian data training dan data testing, kelas tersebut masuk ke dalam data testing maka model akan gagal dalam memprediksi data tersebut.

4. Kesimpulan

Berdasarkan hasil penelitian klasifikasi malware family menggunakan metode Naïve Bayes terhadap dataset CICInvestAndMal2019 setelah melewati tahap feature selection dengan metode Pearson correlation coefficient didapatkan bahwa metode Naïve Bayes dapat melakukan proses klasifikasi malware. Dari keempat komposisi pembagian data testing dan data training, accuracy terbaik mencapai 68% ditunjukkan pada pembagian data 90:10 dengan nilai precision 66% dan Recall 68%. Sedangkan accuracy paling rendah dengan nilai 53% ada pada pembagian data 60:40 dengan nilai precision 61% dan recall 53%.

Penelitian ini juga menunjukan penggunaan metode feature selection menggunakan pearson correlation coefficient serta penerapan K-fold cross validation tidak terlalu berpengaruh dalam meningkatkan nilai accuracy, dimana nilai accuracy tertinggi diperoleh pada fold ke-1 dengan nilai 68% pada dataset tanpa feature selection. Sedangkan dataset yang menggunakan feature selection mendapatkan nilai akurasi 63% pada fold ke-6.

Untuk perbedaan hasil dengan penelitian sebelumnya, random forest mendapatkan hasil lebih baik dengan nilai recall 59,7% dan precision 61% [13], sedangkan Naïve bayes mendapatkan nilai recall 49% dan precision 50%. Namun Naïve Bayes lebih baik jika dibandingkan dengan k-NN dengan nilai recall dan precision 22% [8].

Dari beberapa skenario pengujian yang sudah dijelaskan diatas, disimpulkan bahwa metode Naïve Bayes akan mendapatkan hasil yang baik apabila dataset memiliki komposisi yang sesuai. Maksud dari komposisi yang sesuai disini adalah dalam sebuah kelas seharusnya tidak memiliki 3 kelas saja.

Penelitian selanjutnya diharapkan metode klasifikasi *malware* family dengan *Naïve Bayes* bisa mendapatkan hasil lebih baik dengan memilih dataset dan metode untuk *feature selection* yang sesuai dengan dataset, serta normalisasi dataset yang lebih baik.

Referensi

[1] D. J. Wu, C. H. Mao, T. E. Wei, H. M. Lee, and K. P. Wu, "DroidMat: Android malware detection through manifest and API calls tracing," *Proc. 2012 7th Asia Jt. Conf. Inf. Secur. AsiaJCIS 2012*, pp. 62–69, 2012, doi: 10.1109/AsiaJCIS.2012.18.

[2] A. H. Lashkari, A. F. A. Kadir, L. Taheri, and A. A. Ghorbani, "Toward Developing a Systematic Approach to Generate Benchmark Android Malware Datasets and Classification," *Proc. - Int. Carnahan Conf. Secur. Technol.*, vol. 2018-Octob, no. Cic, pp. 1–7, 2018, doi: 10.1109/CCST.2018.8585560.

[3] Z. Xu, K. Ren, and F. Song, "Android malware family classification and characterization using CFG and DFG," *Proc. - 2019 13th Int. Symp. Theor. Asp. Softw. Eng. TASE 2019*, pp. 49–56, 2019, doi: 10.1109/TASE.2019.00-20.

[4] M. A. Jerlin and K. Marimuthu, "A New Malware Detection System Using Machine Learning Techniques for API Call Sequences," *J. Appl. Secur. Res.*, vol. 13, no. 1, pp. 45–62, 2018, doi: 10.1080/19361610.2018.1387734.

[5] L. Liu, B. sheng Wang, B. Yu, and Q. xi Zhong, "Automatic malware classification and new malware detection using machine learning," *Front. Inf. Technol. Electron. Eng.*, vol. 18, no. 9, pp. 1336–1347, 2017, doi: 10.1631/FITEE.1601325.

[6] L. Massarelli, L. Aniello, C. Ciccotelli, L. Querzoni, D. Ucci, and R. Baldoni, "Android malware family classification based on resource consumption over time," *Proc. 2017 12th Int. Conf. Malicious Unwanted Software, MALWARE 2017*, vol. 2018-Janua, pp. 31–38, 2018, doi: 10.1109/MALWARE.2017.8323954.

[7] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-An, and H. Ye, "Significant Permission Identification for Machine-Learning-Based Android Malware Detection," *IEEE Trans. Ind. Informatics*, vol. 14, no. 7, pp. 3216–3225, 2018, doi: 10.1109/TII.2017.2789219.

[8] A. R. Yogaswara, D. R. Akbi, V. Rahmayati, and S. Nastiti, "Malware Familiy Classification using k-Nearest Neighbor (k-NN)," vol. 3357, no. 1, pp. 1–5, 2020.

[9] H. Zhang, C. T. Liu, J. Mao, C. Shen, R. L. Xie, and B. Mu, "Development of novel in silico prediction model for drug-induced ototoxicity by using naïve Bayes classifier approach," *Toxicol. Vitr.*, vol. 65, no. September 2019, 2020, doi: 10.1016/j.tiv.2020.104812.

[10] L. Taheri, A. F. A. Kadir, and A. H. Lashkari, "Extensible android malware detection and family classification using network-flows and API-calls," *Proc. - Int. Carnahan Conf. Secur. Technol.*, vol. 2019-Octob, no. Cic, 2019, doi: 10.1109/CCST.2019.8888430.

[11] P. Chandrasekar and K. Qian, "The Impact of Data Preprocessing on the Performance of a Naïve Bayes Classifier," *Proc. - Int. Comput. Softw. Appl. Conf.*, vol. 2, pp. 618–619, 2016, doi: 10.1109/COMPSAC.2016.205.

[12] J. D. Chee, "Pearson's Product-Moment Correlation: Sample Analysis," *ResearchGate*, no. May 2015, 2016, doi: 10.13140/RG.2.1.1856.2726.

[13] Z. Zakeri, N. Mansfield, C. Sunderland, and A. Omurtag, "Cross-validating models of continuous data from simulation and experiment by using linear regression and artificial neural networks," *Informatics Med. Unlocked*, vol. 21, no. July, p. 100457, 2020, doi: 10.1016/j.imu.2020.100457.

[14] H. Zhou, Z. Deng, Y. Xia, and M. Fu, "A new sampling method in particle filter based on Pearson correlation coefficient," *Neurocomputing*, vol. 216, pp. 208–215, 2016, doi: 10.1016/j.neucom.2016.07.036.

[15] J. D. Chee and T. Queen, "Pearson's Product Moment Correlation: Sample Analysis," *ResearchGate*, no. May 2015, 2016, doi: 10.13140/RG.2.1.1856.2726.

[16] E. C. Blessie and E. Karthikeyan, "Sigmis: A feature selection algorithm using correlation based method," *J. Algorithms Comput. Technol.*, vol. 6, no. 3, pp. 385–394, 2012, doi: 10.1260/1748-3018.6.3.385.

[17] S. Saud, B. Jamil, Y. Upadhyay, and K. Irshad, "Performance improvement of empirical models for estimation of global solar radiation in India: A k-fold cross-validation approach," *Sustain. Energy Technol. Assessments*, vol. 40, no. June, p. 100768, 2020, doi: 10.1016/j.seta.2020.100768.

[18] G. Jiang and W. Wang, "Error estimation based on variance analysis of k-fold cross-validation," *Pattern Recognit.*, vol. 69, pp. 94–106, 2017, doi: 10.1016/j.patcog.2017.03.025.