Media Journal of General Computer Science, **2024, 1, 7-13**
Published Online January 2024 in MASE (http://www.mase.or.id)
DOI: 10.62205/mjgcs.v1i1.7

Android Malware Detection Using Convolutional Neural Network

Sharipuddin
Department of Informatics, Dinamika Bangsa University, Jambi, Indonesia
E-mail: sharifbuhaira@gmail.com

Rafi Septiandi Putra
Department of Informatics, Dinamika Bangsa University, Jambi, Indonesia
E-mail: rafiseptiandi22@gmail.com

M. Farhan Aulia
Department of Informatics, Dinamika Bangsa University, Jambi, Indonesia
E-mail: farhanaulia64@gmail.com

Sayid Achmad Maulana
Department of Informatics, Dinamika Bangsa University, Jambi, Indonesia
E-mail: syaidahmad7z @gmail.com

Pareza Alam Jusia
Department of Informatics, Dinamika Bangsa University, Jambi, Indonesia
E-mail: parezaalam@gmail.com
*Corresponding Author

Received: 13 November, 2023; Accepted: 23 November, 2023; Published: 30 January, 2024

Abstract: Android is a mobile operating system based on a modified version of the Linux kernel and other open-source tools. Due to its system efficiency and the multitude of features it offers to users, the Android operating system has taken a leading position in the technology market and often attracts the attention of cybercriminals. As malware continues to evolve, traditional methods for detecting Android malware, such as signature-based approaches, may not be sufficient to detect the latest malware threats. Therefore, this research proposes a deep learning algorithm, specifically Convolutional Neural Network (CNN) and Component Analysis (PCA), for feature extraction to enhance the accuracy of Android malware detection. The dataset used in this study is the CICAndMal2017 dataset. Testing results are evaluated using three parameters: accuracy, precision, and recall. Experimental results indicate that our deep learning approach outperforms many other methods with an accuracy of 91%.

Keywords: Malware, Android, PCA, CNN

I. Introduction

Malware adalah kependekan dari *malicious software* dan biasanya digunakan sebagai istilah umum untuk merujuk ke perangkat lunak apa pun yang dirancang untuk menyebabkan kerusakan pada satu komputer, server, jaringan komputer, maupun di Android [1]. Jenis *malware* pun beragam, contohnya *adware*, *Trojan*, *ransomware*, dan lainnya. Tujuan *malware* adalah menyerang file yang dimiliki seseorang dan melakukan duplikasi diri sehingga merusak sistem kerja hardisk dan software, mengambil data, serta merusak sistem operasi pada PC target [2]. Sehingga perlu ada sistem deteksi malware android yang dapat diandalkan.

Signature-Based Detection adalah salah satu teknik *malware detection*, teknik ini menggunakan jejak digital unik, yang dikenal sebagai *signature* atau ciri khas [3]. Perusahaan antivirus menggunakan database besar dari ciri khas malware yang diketahui, biasanya dikelola oleh tim peneliti keamanan yang dioperasikan oleh perusahaan antivirus. Datanya sering diperbarui dan versi terbaru akan disinkronkan dengan perangkat yang dilindungi. Program antivirus memindai perangkat lunak, mengidentifikasi ciri khas nya dan membandingkannya dengan ciri khas malware yang dikenal. Saat program antivirus mengidentifikasi perangkat lunak yang memenuhi ciri khas yang diketahui, antivirus tersebut menghentikan proses dan mengkarantina atau menghapusnya. Ini adalah pendekatan sederhana dan efektif untuk deteksi malware dan penting sebagai garis pertahanan pertama. Namun, karena berkembangnya zaman, penyerang menjadi lebih canggih, Sehingga *Signature-Based Detection* tidak dapat mendeteksi berbagai macam ancaman baru [4][5].

Selama paruh pertama tahun 2022, perusahaan anti virus yang bernama Kaspersky mendeteksi dan memblokir sebanyak 79.442 serangan *malware* yang menargetkan perangkat seluler di Indonesia. Hal ini dibuktikan dengan temuan Kaspersky bahwa banyak aplikasi palsu yang berbeda didistribusikan melalui toko aplikasi resmi. Selain itu, untuk enam bulan pertama tahun 2022, Indonesia berada di peringkat ke-4 secara global dalam hal ancaman seluler.

Masalah yang ada pada sistem deteksi malware android adalah bagaimana meningkatkan kinerja dari sistem deteksi. Beberapa penelitian sebelumnya [6][7][8] telah mengusulkan penelitian untuk meningkatkan kinerja dari deteksi malware di android. Penelitian [6] menggunakan perbandingan algoritma *support vector machine* (SVM) dan *Random Forest* yang digunakan dalam proses klasifikasi terhadap *dataset malware*. Pada pengujian didapatkan hasil bahwa metode *Random Forest* lebih unggul dibandingkan dengan metode SVM untuk kasus ini. Dengan hasil akurasi *Random Forest* sampai 98,99% sedangkan metode SVM mendapatkan hasil akurasi sampai 96,23%. Selanjutnya Penelitian [7] menggunakan *Artificial Neural Network* (ANN) dan arsitektur LSTM, hasil pengujian model menggunakan data uji menunjukkan akurasi dan skor F1 sebesar 98,7%. Kemudain pada [8] menggunakan *Gated Recurrent Unit* (GRU) yaitu salah satu tipe dari *Recurrent Neural Network* (RNN) dan *dataset* dari *CICAndMal2017*. Hasil percobaan menunjukkan bahwa metode nya mengungguli beberapa metode lainnya. Berdasarkan dari penelitian sebelumnya metode yang termasuk ke *deep learning* menjadi metode yang menjanjikan untuk mendeteksi malware android.

Oleh karena itu, pada penelitian ini memanfaatkan keunggulan *deep learning* untuk mencapai deteksi akurasi *malware* yang lebih baik. *Deep Learning* (DL) merupakan sub bagian dari *machine learning*, yang fokus pada pengembangan sebuah sistem yang mampu belajar sendiri tanpa perlu berulang kali di program oleh manusia[9][10]. Penelitian ini mengusulkan metode DL untuk mendeteksi *malware* Android menggunakan metode *Convolutional Neural Network* (CNN) dan *Principal Components Analysis* (PCA) untuk *feature extraction* nya.

Adapun penjabaran pada makalah ini adalah sebagai berikut. Bagian 2 menyajikan penelitian-penelitian sejenis. Kemudian penjelasan terkait metodologi di bagian 3 dan dilanjutkan pada bagian 4 yang menunjukkan hasil eksperimen beserta pembahasannya. Pada bagian akhir yakni bagian 5, kami menutup dengan kesimpulan penelitian ini.

2. Research Method

2.1. Experiment Setup

Penelitian ini bertujuan untuk mengusulkan metode CNN untuk mendeteksi *malware* Android. Ada beberapa tahapan penelitian yang harus dilakukan untuk dapat mencapai tujuan dari penelitian ini, maka perlu dirancang alur penelitian yang ditampilkan pada Fig 1.

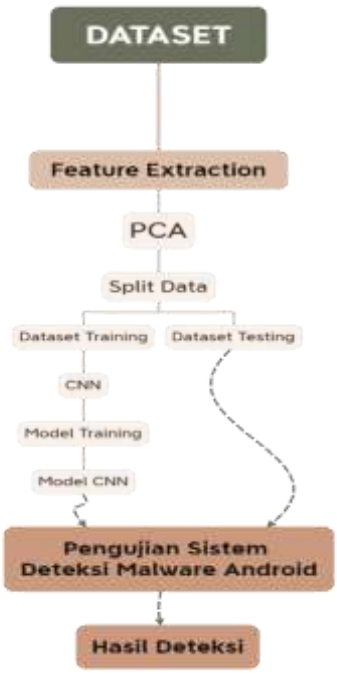

Fig.1. Konfigurasi Eksperimen

Fig 1 adalah konfigurasi eksperimen yang di desain pada penelitian ini. Oleh karena itu pada penelitian ini dapat dibagi menjadi tiga tahapan yaitu:

- Melakukan ekstraksi fitur dari *dataset* menggunakan metode PCA serta membagi *dataset* untuk data *training* dan data *testing.*
- Selanjutnya adalah melakukan *training* model CNN untuk deteksi *malware* menggunakan *dataset training* sehingga diperoleh model CNN yang akan digunakan untuk proses pengujian.
- Terakhir, melakukan pengujian menggunakan model CNN dengan diujikan menggunakan *dataset training* serta menghitung keberhasilan seperti akurasi, presisi dan *recall.*

2.2. Dataset

Penelitian ini akan menggunakan CIC-AndMal2017 *dataset* yang digunakan untuk deteksi aktivitas jahat atau *malware* pada jaringan komputer. *Dataset* ini dikembangkan oleh *Canadian Institute Cybersecurity* (CIC) dan terdiri dari bagian normal dan bagian *malware*. Bagian normal berisi lalu lintas jaringan yang sah, sedangkan bagian *malware* berisi lalu lintas jaringan yang berasal dari aktivitas *malware*. Dalam kumpulan data ini, terdapat lebih dari 10,854 *samples* (4.354 *malware* dan 6.500 tidak berbahaya), sampel *malware* dalam *dataset* CICAndMal2017 diklasifikasikan ke dalam empat kategori *Adware*, *Ransomware*, *Scareware*, dan SMS *Malware*. *Dataset* CIC-AndMal2017 terdiri dari beberapa sub-*dataset* yang mencakup berbagai skenario dan jenis serangan [11].

2.3. Feature Extraction Menggunakan PCA

Feature extraction bertujuan untuk mengekstraksi fitur dari fitur asli yang sudah ada dan memodifikasi fitur pada ukuran yang lebih kecil untuk mempercepat proses pelatihan dan meningkatkan akurasi. Pada penelitian mengusulkan *Principal Components Analysis* (PCA) untuk mengurangi dimensi dari *dataset*. Detail dari penggunaan *pseudocode PCA* pada penelitian ini ditampilkan di bawah ini. Penelitian ini akan mereduksi dimensi menjadi 5, 8, 10 fitur dan akan digunakan pada proses *training* dan deteksi seperti pada tabel 1.

Tabel 1. Pseudocode PCA

Pseudocode PCA
sklearn.decomposition import PCA
dataset ← load dataset
(x_train, y_train),(x_test, y_test) ← dataset_ CIC-AndMal2017
pca ←PCA(n_componnets=5, 8, 10)
x_pca ← pca.fit_transform((x_train, y_train),(x_test, y_test))

2.4. Proposed Method

Convolutional Neural Network (CNN) adalah jenis arsitektur jaringan saraf tiruan yang sangat efektif dalam memproses data berstruktur *grid*, seperti citra atau data spasial. CNN dirancang khusus untuk menangani data dengan pola lokal yang kuat, sehingga sangat cocok untuk aplikasi pengolahan citra dan pengenalan pola.CNN terdiri dari berbagai lapisan yang dirancang untuk melakukan ekstraksi fitur secara hierarkis. Arsitektur dari CNN ditampilkan pada Fig 2.

Lapisan pertama dalam CNN adalah lapisan konvolusi, yang menggunakan filter atau kernel untuk menerapkan operasi konvolusi pada input. Filter ini bergerak melintasi input untuk mengekstraksi fitur-fitur lokal. Lapisan konvolusi dapat mendeteksi pola seperti tepi, garis, atau bentuk yang lebih kompleks dalam data.

Setelah lapisan konvolusi, biasanya diikuti oleh lapisan pengecilan (*pooling layer*), yang bertujuan untuk mereduksi dimensi data dan mengurangi jumlah parameter yang dibutuhkan oleh model. Lapisan ini dapat melakukan operasi seperti *max pooling*, yang memilih nilai maksimum dalam jendela atau wilayah tertentu.

Setelah beberapa lapisan konvolusi dan pengecilan, lapisan-lapisan *fully connected* digunakan untuk klasifikasi akhir. Lapisan-lapisan ini menghubungkan setiap fitur yang terekstraksi ke setiap kelas yang mungkin, dan *output* nya berupa probabilitas untuk setiap kelas.

Proses pelatihan CNN melibatkan pemberian contoh data dengan label ke dalam jaringan. Selama pelatihan, bobot dan bias dalam jaringan disesuaikan menggunakan algoritma optimasi seperti *backpropagation* dan *stochastic gradient descent* (SGD), untuk mengurangi selisih antara output yang dihasilkan oleh jaringan dan label yang sebenarnya.

CNN telah terbukti sangat efektif dalam berbagai aplikasi, termasuk pengenalan objek dalam citra, deteksi wajah, pengolahan bahasa alami, dan juga deteksi *malware*. Dalam konteks deteksi *malware* di Android, CNN dapat digunakan untuk mengidentifikasi pola-pola yang mencirikan aplikasi *malware*, seperti *opcode*, penggunaan sumber daya, atau perilaku aplikasi yang mencurigakan.

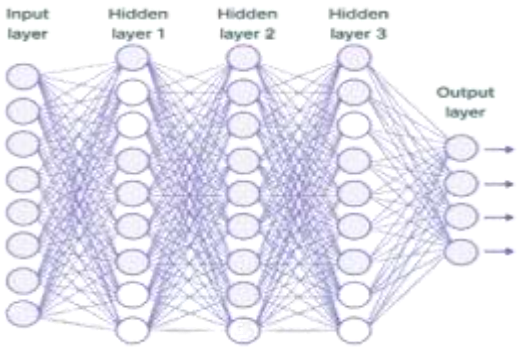

Fig.2. Algoritma CNN

2.4. Environment Setup

Semua *experiment* di dalam percobaan ini dilakukan pada komputer yang menjalankan sistem operasi *Windows* 11, dengan spesifikasi prosesor AMD Ryzen 5 2,00 GHz dan RAM 16 GB. Untuk keperluan analisis, digunakan python dengan perangkat lunak seleksi fitur, *scikit-learning*, tensorflow dan Keras untuk CNN

3. Results and Discussion

Bagian ini akan membahas hasil eksperimen yang di uji coba. Pembahasan terdiri dari hasil *feature extraction* dan hasil pengujian sistem deteksi menggunakan CNN

3.1. Hasil PCA

Langkah pertama adalah mengonversi *dataset* ke dimensi yang lebih kecil. Tujuannya untuk mengurangi beban komputasi dan meningkatkan performa sistem deteksi *malware* di Android. Penelitian ini menggunakan metode PCA untuk mereduksi dimensi atribut *dataset* tanpa menghilangkan karakteristik datanya. Tabel 2 menunjukkan hasil ekstraksi fitur yang dilakukan dengan metode PCA. Penelitian ini mereduksi dataset menjadi 5, 8, dan 10 atribut. Hasil *feature extraction* ini digunakan untuk mengolah data *training* menggunakan CNN. Pada proses PCA, *record* diubah menjadi nilai dengan *range* 0 sampai 1. Setelah dikonversi, nilai dari fitur menjadi *range* yang lebih kecil dengan tujuan untuk mengurangi sumber daya *dataset* menggunakan metode CNN

Tabel 2. Hasil pemilihan fitur

Jumlah fitur	Hasil PCA
5	234565,-7, 9731.4, 24450.34, -13594, 142.36032
8	456565 -3, 9731.7,24450.34, -13594, 536.2613, -5.33322,-0.15642, 5.175025
10	161545 -3, 9721.7, 22450.34, -43564, 848.3613,-5.99922,-0.23241, 4.045035, 2.30423, 1.922208,

3.2. Hasil Deteksi CNN

Tabel 3 adalah hasil pengujian CNN pada *dataset* CIC-AndMal2017. Hasil pengujian terdiri dari 3 parameter yaitu akurasi, presisi dan *recall*. Hasil pengujian memperoleh akurasi yang cukup memuaskan untuk beberapa *dataset*. Hasil akurasi terbaik diperoleh pada *dataset PornDroid* dan *Charger* yang memiliki selisih hasil yang hampir serupa yaitu 91.3% dan 91%. Hasil rata-rata akurasi pada pengujian deteksi menggunakan CNN pada *dataset* CIC-AndMal2017 mencapai 83.48%. Selain itu pada parameter presisi dan recall memperoleh hasil yang cukup memuaskan. Hasil yang kurang memuaskan diperoleh pada *dataset Pletor* yang hanya mencapai 55% serta pada *dataset Lockerpin* yang hanya mencapai 70.8%.

Tabel 3. Hasil Pengujian CNN

Nama Dataset	**Akurasi**	**Presisi**	**Recall**
Charger	0.910	0.910	0.910
Jisut	0.876	0.881	0.890
Koler	0.883	0.885	0.897
Lockerpin	0.708	0.712	0.714
Pletor	0.550	0.582	0.617

PornDroid	0.913	0.913	0.915
RansomBO	0.876	0.883	0.902
Simplocker	0.884	0.887	0.889
SVpeng	0.893	0.896	0.894
Wanna	0.855	0.859	0.858

Fig.3. Hasil Pengujian Akurasi

Fig.4. Hasil Pengujian Presisi

Fig.5. Hasil Pengujian Recall

Fig 3 adalah hasil pengujian deteksi *malware* pada Android menggunakan CNN yaitu akurasi. Hasil akurasi, presisi dan *recall* menunjukkan performa yang hampir mirip dari pola grafiknya yang mencapai 91 sampai 85 persen. Fig 4 dan Fig 5 adalah hasil pengujian deteksi *malware* pada Android menggunakan CNN yaitu presisi dan *recall*. Terlihat pada grafik diatas bahwa algoritma CNN menunjukkan hasil performa yang cukup stabil untuk setiap *dataset* akan tetapi masih dalam ambang batas yang bagus. Dari pengujian ini dapat disimpulkan bahwa CNN berhasil mendeteksi serangan *malware* yang terjadi pada Android.

4. Conclusion

Seiring berkembangnya teknologi mobile malware pun ikut berkembang, metode deteksi malware Android tradisional mungkin tidak akan cukup untuk mendeteksi malware terbaru. Tujuan dari penelitian ini adalah

meningkatkan performa dari sistem deteksi *malware* Android menggunakan CNN dan PCA. Tiga parameter pengukuran yang digunakan dalam penelitian ini yaitu akurasi, presisi dan *recall*. Berdasarkan hasil perhitungan akurasi, presisi dan *recall*, performa dari CNN dengan *feature extraction* (PCA) berhasil dalam mendeteksi serangan *malware* pada Android dengan akurasi mencapai 83.48%. Studi ini menyarankan untuk menggunakan metode DL yang lain seperti RNN dan fitur seleksi lainnya di masa mendatang untuk mendeteksi serangan *malware* pada Android lebih baik lagi

Acknowledgment

Penelitian ini didukung oleh Universitas Dinamika Bangsa, Jambi, Indonesia

References

[1] M. K. Alzaylaee, S. Y. Yerima, and S. Sezer, "DL-Droid: Deep learning based android malware detection using real devices," *Comput. Secur.*, vol. 89, p. 101663, 2020, doi: 10.1016/j.cose.2019.101663.
[2] X. Wang and C. Li, "Android malware detection through machine learning on kernel task structures," *Neurocomputing*, vol. 435, pp. 126–150, 2021, doi: 10.1016/j.neucom.2020.12.088.
[3] S. Lee *et al.*, "LARGen: Automatic Signature Generation for Malwares Using Latent Dirichlet Allocation," *IEEE Trans. Dependable Secur. Comput.*, vol. 15, no. 5, pp. 771–783, 2018, doi: 10.1109/TDSC.2016.2609907.
[4] A. Souri and R. Hosseini, "A state-of-the-art survey of malware detection approaches using data mining techniques," *Human-centric Comput. Inf. Sci.*, vol. 8, no. 1, 2018, doi: 10.1186/s13673-018-0125-x.
[5] H. M. Deylami, R. C. Muniyandi, I. T. Ardekani, and A. Sarrafzadeh, "Taxonomy of malware detection techniques: A systematic literature review," *2016 14th Annu. Conf. Privacy, Secur. Trust. PST 2016*, pp. 629–636, 2016, doi: 10.1109/PST.2016.7906998.
[6] Y. Wanli Sitorus, P. Sukarno, dan S. Mandala, "Analisis Deteksi Malware Android menggunakan metode Support Vector Machine & Random Forest," *e-Proceeding of Engineering*, vol. 8, no. 6, hlm. 12500, 2021.
[7] R. B. Hadiprakoso, N. Qomariasih, dan R. N. Yasa, "IDENTIFIKASI MALWARE ANDROID MENGGUNAKAN PENDEKATAN ANALISIS HIBRID DENGAN DEEP LEARNING," *Jurnal Teknologi Informasi Universitas Lambung Mangkurat*, vol. 6, no. 2, hlm. 77–84, 2021.
[8] O. N. Elayan dan A. M. Mustafa, "Android malware detection using deep learning," dalam *Procedia Computer Science*, Elsevier B.V., 2021, hlm. 847–852. doi: 10.1016/j.procs.2021.03.106.
[9] Sharipuddin, E. A. Winanto, Z. Z. Mohtar, Kurniabudi, I. S. Wijaya, and D. Sandra, "Improvement detection system on complex network using hybrid deep belief network and selection features," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 31, no. 1, pp. 470–479, 2023, doi: 10.11591/ijeecs.v31.i1.pp470-479.
[10] S. Sharipuddin *et al.*, "Enhanced Deep Learning Intrusion Detection in IoT Heterogeneous Network with Feature Extraction," *Int. J. Electr. Eng. Informatics*, vol. 9, no. 3, pp. 747–755, 2021, doi: 10.52549/ijeei.v9i3.3134.
[11] 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-October, no. Cic, pp. 1–7, 2018, doi: 10.1109/CCST.2018.8585560.

Authors' Profiles

Dr. Sharipuddin menerima gelar Doktor Teknik dari Universitas Sriwijaya. Saat ini dia menjabat sebagai Dosen Senior di Fakultas Ilmu Komputer, Universitas Dinamika Bangsa, Indonesia. Minat penelitiannya meliputi teknologi informasi dan keamanan informasi

Rafi Septiandi Putra lahir di Jambi, Indonesia. Saat ini dia tengah menempuh Program Sarjanah Informatika di Universitas Dinamika Bangsa, Indonesia. Dia memiliki fokus penelitian pada machine learning dan software engineering.

M. Farhan Aulia lahir di Jambi, Indonesia. Saat ini dia tengah menempuh Program Sarjanah Informatika di Universitas Dinamika Bangsa, Indonesia. Dia memiliki fokus penelitian pada machine learning dan software engineering.

Sayid Achmad Maulana lahir di Jambi, Indonesia. Saat ini dia tengah menempuh Program Sarjanah Informatika di Universitas Dinamika Bangsa, Indonesia. Dia memiliki fokus penelitian pada machine learning dan software engineering

Pareza Alam Jusia meraih gelar Sarjana di bidang Teknik Informatika dari Universitas Dinamika Bangsa Jambi, dan gelar M.Kom di bidang Rekayasa Perangkat Lunak dari Universitas Budi Luhur Jakarta, Indonesia. Saat ini beliau adalah dosen di Fakultas Ilmu Komputer, Universitas Dinamika Bangsa, Indonesia. Minat penelitiannya meliputi Machine Learning, Decision Support System dan Artificial Intelligence for Bussiness.