Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 5 No. 3 Juli 2023 Hal. 217-223
http://jurnal.unidha.ac.id/index.php/jteksis

E-ISSN : **2655-8238**
P-ISSN : **2964-2132**

Optimasi Algoritma Random Forest menggunakan Principal Component Analysis untuk Deteksi Malware

Fauzi Adi Rafrastara[a*], Ricardus Anggi Pramunendar[b], Dwi Puji Prabowo[c], Etika Kartikadarma[d], Usman Sudibyo[e]

[a,b,c,d,e]Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang
[a]fauziadi@dsn.dinus.ac.id, [b]ricardus.anggi@dsn.dinus.ac.i.d, [c]dwi.puji.prabowo@dsn.dinus.ac.id, [d]etika.kartikadarma@dsn.dinus.ac.id, [e]usman.sudibyo@dsn.dinus.ac.id

Submitted: 07-06-2023, Reviewed: 15-06-2023, Accepted 30-06-2023
https://doi.org/10.47233/jteksis.v5i3.854

Abstract
Malware is a type of software designed to harm various devices. As malware evolves and diversifies, traditional signature-based detection methods have become less effective against advanced types such as polymorphic, metamorphic, and oligomorphic malware. To address this problem, machine learning-based malware detection has emerged as a promising solution. Detection using machine learning also has its own challenges. One of them is about how to optimize the performance of machine learning algorithms through feature reduction, considering that the features generated during malware behavior analysis are very large, reaching 1087 features. With such a large number of features, it will certainly have an impact on the performance of machine learning algorithms, in terms of accuracy, recall, and computation time. In this study, we evaluated the performance of several machine learning algorithms in detecting malware and applied Principal Component Analysis (PCA) to the best-performing algorithm to reduce the number of features and improve performance. Our results showed that the Random Forest algorithm outperformed Adaboost, Neural Network, Support Vector Machine, and k-Nearest Neighbor algorithms with an accuracy and recall rate of 98.3%. By applying PCA, we were able to further improve the performance of Random Forest to 98.7% for both accuracy and recall while reducing the number of features from 1084 to 32.
Keywords: *Random forest, principal component analysis, features reduction, malware detection*

Abstrak
Malware merupakan software yang memiliki perangai jahat dan dapat menyerang beragam perangkat. Evolusi malware pun terus terjadi dengan jenis yang semakin beragam. Deteksi malware berbasis signature sudah mulai ditinggalkan karena ketidak-efektifannya dalam menangani malware jenis polymorphic, metamorphic hingga oligomorphic. Oleh karena itu, deteksi malware berbasis machine learning menjadi sangat penting untuk dilakukan. Deteksi menggunakan machine learning pun memiliki tantangan tersendiri. Salah satunya yaitu tentang bagaimana mengoptimasi performa algoritma machine learning melalui reduksi fitur, mengingat fitur-fitur yang dihasilkan pada saat analisis perilaku malware sangatlah besar, yaitu mencapai 1087 fitur. Dengan jumlah fitur yang sedemikian banyak ini tentu akan berdampak ke performa algoritma machine learning, baik dari sisi akurasi, recall, maupun waktu komputasi. Tujuan dari penelitian ini adalah untuk mengetahui performa beberapa algoritma machine learning dalam mendeteksi malware, kemudian menerapkan algoritma reduksi fitur Principal Component Analysis (PCA) pada algoritma terbaik guna meningkatkan performanya. Sebagai hasil, Random Forest menjadi algoritma yang paling unggul jika dibandingkan dengan 4 algoritma lain, seperti Adaboost, Neural Network, Support Vector Machine, dan k-Nearest Neighbor, dimana nilai akurasi dan Recall yang diperoleh adalah sebesar 98.3%. Selanjutnya, dengan PCA, performa Random Forest berhasil ditingkatkan menjadi 98.7%, baik untuk skor akurasi maupun recall, meskipun jumlah fiturnya direduksi menjadi 32 (semula 1084 fitur).
Keywords: Random forest, principal component analysis, reduksi fitur, deteksi malware

PENDAHULUAN

Malware merupakan software yang memiliki perangai malicious atau jahat dan dapat menyerang beragam perangkat, mulai dari komputer PC, laptop, tablet hingga smartphone [1]–[3]. Ada beberapa jenis malware yang populer, seperti: virus, trojan horse, worm, spyware, botnet, hingga ransomware [4]. Masing-masing malware tersebut memiliki pola perilaku dan tujuan yang berbeda-beda. Malware dengan tujuan utama untuk merusak atau menginfeksi komputer korban, dikenal dengan nama Virus [5], [6]. Begitu virus muncul di dunia komputer, pengembangan anti virus pun langsung dilakukan. Eksistensi virus terancam berkat adanya anti virus. Oleh karena itu, virus sederhana kemudian berevolusi menjadi stealth virus yang mana memiliki kemampuan untuk bersembunyi dari deteksi antivirus [7].

Berikutnya, terdapat suatu malware yang dapat menyebarkan diri secara massive layaknya cacing dengan membawa misi tertentu sesuai keingingan pembuatnya. Malware ini dikenal dengan nama Worm [8]. Di sisi lain, ada pula malware berjenis Botnet, yaitu malware yang dapat dikendalikan dari

Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 5 No. 3 Juli 2023 Hal. 217-223
http://jurnal.unidha.ac.id/index.php/jteksis

E-ISSN : **2655-8238**
P-ISSN : **2964-2132**

jarak jauh oleh pembuatnya untuk menyusup ke sistem target dan mengeksploitasi secara diam-diam[9], [10].

Sedangkan malware yang akhir-akhir ini menjadi masalah bagi banyak orang dan juga dunia industri, bernama Ransomware. Ransomware memiliki sifat destruktif layaknya virus disertai dengan permintaan tebusan melalui bitcoin [4]. Tebusan ini digunakan untuk mengembalikan data atau system ke kondisi awal, mengingat ransomware memiliki ciri khas yaitu mengengkrip file-file tertentu atau seluruh file pada komputer target.

Malware berkembang bukan saja berdasarkan jenisnya, namun juga kemampuannya dalam menghindari deteksi. Antivirus konvensional yang berbasis signature-based detection hanya bisa bekerja pada virus berjenis monomorphic atau tradisional. Pada virus berjenis polymorphic dimana induk suatu malware dapat beranak pinak dengan menghasilkan signature yang berbeda-beda, tentu konvensional antivirus tidak akan mampu secara efektif menanganinya [5]. Polymorphic malware bisa secara acak menggenerate signature untuk file baru (anakan malware) dan tentu ini aka menyulitkan antivirus karena harus menyimpan semakin banyak signature malware pada databasenya. Dengan demikian, penanganan malware menjadi sangat tidak efektif. Sebagai solusi, dibutuhkan smart sistem yang memiliki kemampuan menganalisis dan mendeteksi suatu malware, baik secara static maupun dynamic.

Pada penelitian ini, dataset malware digunakan untuk membandingkan 5 algoritma machine learning (Random Forest, AdaBoost, Neural Network, Support Vector Machine (SVM), dan k-Nearest Neighbor (kNN)) guna mendapatkan algoritma terbaik. Untuk lebih mengoptimalkan algoritma tersebut, metode Principal Component Analysis dimanfaatkan untuk mereduksi fitur pada dataset yang diperoleh dari UCI Machine Learning Repository. Performa algoritma dengan PCA akan diukur untuk mengetahui peningkatan antara sebelum dan setelah ditambah PCA untuk reduksi fitur. Performa yang diukur adalah terkait dengan akurasi dan recall.

Random Forest merupakan pengembangan dari algoritma Decision Tree berbasis ensemble yang dapat digunakan pada kasus klasifikasi maupun regresi. Klasifikasi merupakan upaya menemukan atau membangun sebuah model dengan data kelas berupa nilai diskrit. Sementara itu, regresi digunakan ketika data kelas berupa nilai kontinyu. Penelitian ini fokus pada klasifikasi, khususnya mengklasifikasi suatu file apakah tergolong sebagai malware atau goodware. Beberapa contoh algoritma klasifikasi yang populer adalah Naive Bayes, KNN, C4.5, hingga Random Forest [11]–[16]. Mengingat Random Forest merupakan algoritma ensemble, Random Forest memiliki performa yang lebih baik dibandingkan algoritma-algoritma berbasis tree lainnya. Random Forest tergolong algoritma yang sukses dan telah diterapkan secara luas baik di dunia akademik maupun industri [13].

Pada bidang information security, algoritma Random Forest juga banyak digunakan untuk melakukan klasifikasi malware. Peneliti pada [14] menggunakan Random Forest untuk mengklasifikasi malware. Dataset yang digunakan ialah Malimg Dataset yang terdiri dari 9.342 sample malware, dengan kelas yang tidak seimbang (imbalanced dataset). Untuk mencegah terjadinya overfitting, peneliti menggunakan metode stratified sampling. Hasilnya, akurasi yang diperoleh adalah sebesar 95.62%.

Artikel [15] juga mengusung algoritma berbasis Random Forest untuk mendeteksi malware pada platform Android. Dataset yang digunakan bernama Hemdds serta terdiri dari 1065 goodware dan 1065 malware (balanced dataset). Hasilnya, akurasi yang diperoleh menggunakan Random Forest ialah sebesar 89.91%.

Khammas [16], pada penelitiannya menerapkan algoritma Random Forest untuk mendeteksi Ransomware. 1680 executable files dianalisis, dimana 840 file masuk ke dalam kelas Ransomware dan 840 sisanya adalah goodware (balanced dataset). Peneliti menerapkan feature selection sehingga jumlah fiturnya diturunkan dari 7000 fitur menjadi 1000 fitur. Hasil eksperimen dengan 1000 fitur menghasilkan performa akurasi yang terbaik, yaitu 97.74%.

Berangkat dari penelitian-penelitian di atas, khususnya oleh Khammas [16], dibutuhkan metode reduksi fitur yang lain untuk meningkatkan performa algoritma Random Forest. Pada penelitian ini, algoritma PCA akan diterapkan untuk mereduksi fitur pada dataset sekaligus untuk meningkatkan performa algoritma Random Forest, khususnya dari sisi akurasi dan recall. Dengan demikian, maka deteksi malware menggunakan algoritma Random Forest akan dapat berlangsung dengan lebih efektif mengingat adanya reduksi jumlah fitur secara signifikan, sehingga tidak hanya nilai akurasi dan recallnya saja yang meningkat, namun juga terjadi peningkatan pada kecepatan prosesnya.

METODE PENELITIAN

Pada penelitian ini, akan dilaksanakan 2 jenis eksperimen (Gambar 1). Eksperimen pertama dilakukan untuk memperoleh algoritma dengan performa terbaik (algoritma X) dalam mendeteksi malware di antara 5 algoritma klasifikasi, seperti: Random Forest, Adaboost, Neural Network, Support Vector Machine, dan k-Nearest Neighbor.

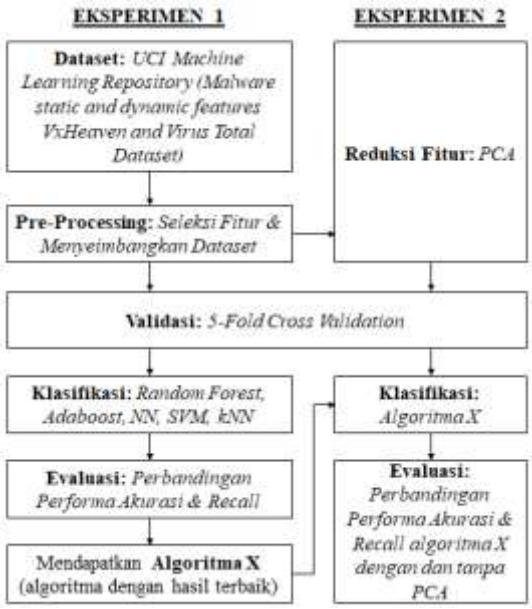
Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 5 No. 3 Juli 2023 Hal. 217-223
http://jurnal.unidha.ac.id/index.php/jteksis

E-ISSN : **2655-8238**
P-ISSN : **2964-2132**

Sedangkan pada eksperimen ke dua, reduksi fitur dengan Principal Component Analysis akan diterapkan guna meningkatkan performa algoritma X. Bab ini akan membahas tahapan-tahapan penelitian dan juga eksperimen, dimulai dari pembahasan tentang hardware dan software yang digunakan, dataset, pre-processing, validasi, hingga evaluasi.

2.1. Hardware dan Software

Dalam penelitian terkait bidang ilmu komputer, dukungan perangkat keras maupun lunak memainkan peranan yang sangat penting dalam keberhasilan suatu penelitian. Software yang baik tanpa didukung hardware yang mumpuni, tidak akan bisa berjalan secara optimal. Sebaliknya, meskipun hardwarenya memiliki spesifikasi yang tinggi, namun tidak diimbangi denga software yang tepat, maka akan menjadi sia-sia. Oleh karena itu, penggunaan hardware dan software yang tepat menjadi sangat penting guna menunjang keberhasilan penelitian.



Gambar 1. Alur eksperimen 1 dan 2

Pada penellitian ini, eksperimen dilakukan di komputer dengan spesifikasi sebagai berikut:

A. Processor : Intel Xeon E5620
B. RAM : 16GB
C. HD : 3TB
D. VGA : Radeon RX550

Adapun terkait software, sebuah aplikasi desktop bernama Orange (https://orangedatamining.com/ digunakan untuk memproses dataset.

2.2. Dataset

Data yang digunakan pada penelitian ini diperoleh dari UCI Machine Learning Repository, dengan detail seperti pada Tabel 1.

Tabel 1. Detail dataset yang digunakan

Nama Dataset	Malware static and dynamic features VxHeaven and Virus Total Data Set.
Jumlah file	3 (terdiri dari file goodware, malware dari VirusTotal, dan malware dari VxHeaven).
Jumlah baris	Goodware: 595; VirusTotal: 2955; VxHeaven: 2698
Jumlah fitur	Goodware: 1085; VirusTotal: 1087; VxHeaven: 1087 (selain label).
Missing value	Tidak ada

Di dalam dataset yang didownload tersebut terdapat 3 buah file yang terdiri dari file goodware, file malware dari VirusTotal, dan file malware dari VxHeaven. File goodware berisi rekaman aktivitas 595 file yang tidak terindikasi sebagai virus. 595 file tersebut dieksekusi pada suatu sandbox dan hasil rekaman aktifitasnya menghasilkan 1085 fitur.

Pada file yang kedua, terdapat hasil rekaman dari 2955 file virus yang dieksekusi di dalam sandbox dan menghasilkan data-data dengan 1087 fitur. Virus-virus yang dicatat pada file kedua ini diperoleh dari VirusTotal. Sedangkan file yang ketiga merupakan hasil rekaman perilaku 2698 file virus dari VxHeaven. Adapun fitur yang dihasilkan berjumlah 1087 fitur.

Pada tahap ini, dua file malware akan digabung dan diberi label ‘0’ untuk mengindikasikan sebagai malware. Sementara hasil rekaman pada file goodware ditambahkan label ‘1’ di setiap datanya. Pada akhirnya data-data berlabel goodware dan malware akan dijadikan satu ke dalam sebuah dataset tunggal sebelum diproses lebih lanjut dengan machine learning. Pada akhirnya, terbentuklah sebuah dataset yang berisi data-data malware dan goodware dengan jumlah total data mencapai 6248 data.

2.3. Pre-Processing

Data yang telah dikumpulkan perlu diolah terlebih dahulu sebelum diproses dengan machine learning. Jumlah fitur yang pada data-data berlabel goodware dan malware memiliki perbeedaan. Data-data berlabel goodware memiliki 1085 fitur, sementara yang berlabel malware memiliki 1087 fitur. Terdapat selisih 2 fitur yang setelah diselidiki, ternyata 2 fitur tambahan yang ada pada data-data berlabel malware mengandung nilai 0 di seluruh datanya. Kedua fitur tersebut bernama vbaVarIndexLoad dan SafeArrayPtrOfIndex. Mengingat keduanya hanya memiliki nilai 0 di seluruh data, maka 2 fitur tersebut akan dihapus supaya data-data berlabel malware maupun goodware memiliki identitas fitur yang sama persis,

Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 5 No. 3 Juli 2023 Hal. 217-223
http://jurnal.unidha.ac.id/index.php/jteksis

E-ISSN : **2655-8238**
P-ISSN : **2964-2132**

baik nama maupun jumlahnya. Selain 2 fitur tersebut, fitur 'filename' juga dihapus. Dengan demikian, tersisa 1084 fitur, baik untuk goodware maupun malware.

Di sisi yang lain, jumlah data pada malware dan goodware juga berbeda. Berdasarkan data yang didownload dari UCI, terdapat 595 data goodware dan 5653 data malware. Rasio yang diperoleh dari dua data tersebut adalah 1:9.5. Rasio demikian dinilai cukup tinggi sehingga dapat berpengaruh pada performa algoritma machine learning. Kasus demikian disebut dengan imbalanced dataset. Dengan adanya masalah ini, maka dataset tersebut perlu diseimbangkan terlebih dahulu menggunakan metode Random Under Sampling. Dengan demikian, akan diperoleh jumlah data goodware dan malware yang seimbang, yaitu masing-masing 595 data, sehingga total berjumlah 1190 data.

2.4 Reduksi Fitur dengan Principal Component Analysis

Dengan 1084 fitur tentu akan berdampak pada processing time. Salah satu cara untuk mempercepat processing time adalah dengan mengurangi jumlah fiturnya, namun tetap menjaga kualitas dataset. Dan salah satu algoritma reduksi fitur yang populer dan dapat diandalkan untuk melakukan itu adalah Principan Componnent Analysis (PCA) [17]–[19]. Adapun tahapan PCA adalah sebagai berikut:

1. Menghitung matriks kovarian dengan menggunakan persamaan 1.

$$Cov(xy) = \frac{\sum xy}{n} - (\bar{x})(\bar{y}) \quad (1)$$

2. Menghitung nilai eigen sesuai dengan persamaan 2.

$$(A - \lambda I) = 0 \quad (2)$$

3. Menghitung vektor eigen dengan menggunakan Persamaan 3.

$$[A - \lambda I][X] = [0] \quad (3)$$

4. Menentukan variabel baru (principal component) dengan mengalikan variabel asli dengan matriks vektor eigen.

2.5. Pemodelan

Pada tahap pemodelan, dua jenis eksperimen dilakukan pada penelitian ini. Eksperimen yang pertama adalah dengan membandingkan peforma 5 algoritma klasifikasi dalam mendeteksi malware. Performa yang diukur adalah akurasi dan juga recall mengingat Fale Negatif pada kasus deteksi virus memiliki resiko yang tinggi. Adapun algoritma yang dibandingkan pada tahap ini adalah Random Forest, Adaboost, Neural Network, Support Vector Machine (SVM), dan k-Nearest Neighbor (kNN).

Selanjutnya, setelah diperoleh algoritma dengan performa terbaik, maka hasil performa dari algoritma tersebut akan ditingkatkan dengan menerapkan metode Principal Componnent Analysis (PCA) untuk reduksi jumlah fiturnya. 10 principle components dengan variance di atas 80% akan diuji nilai akurasi dan recallnya. Standar tersebut merupakan salah satu cara dalam memilih jumlah PC, selain dengan mengamati nilai eigen juga patahan siku pada screeplot [18], [20], [21]. Pada akhirnya, jumlah principle component paling sedikit (dengan variance di atas 80%) dan memiliki nilai akurasi serta recall tertinggi-lah yang akan dipilih.

2.6. Evaluasi

Metrik evaluasi pada kasus klasifikasi yang digunakan pada penelitian ini adalah akurasi dan recall. Akurasi berguna untuk memberikan gambaran seberapa sering suatu algoritma machine learning memprediksi dengan benar. Hasil perhitungan akurasi berupa prosentase ketepatan prediksi oleh suatu model. Adapun formula yang digunakan untuk menghitung akurasi dapat dilihat pada persamaan 4. TP atau True Positive mewakili jumlah prediksi yang tepat untuk kelas positive. TN atau True Negative berisi jumlah prediksi yang tepat untuk kelas negative. Sementara itu, FP atau False Positive merupakan banyaknya kesalahan model dalam memprediksi positive pada kelas aktual negative. Sebaliknya, FN atau False Negative merupakan banyaknya kesalahan yang dilakukan oleh model machine learning dalam memprediksi nilai negative pada kelas aktual positive.

$$Accuracy = \frac{TP+TN}{(TP+FP+TN+ FN)} \quad (4)$$

Metrik kedua yang digunakan adalah Recall atau Sensitivitiy. Istilah lain yang juga dikenal adalah TPF atau True Positive Rate. Metrik ini berguna untuk menghasilkan pengukuran terhadap prediksi positive pada kelas aktual positive. Dalam kasus deteksi malware, Recall memiliki peranan yang sangat penting, mengingat peristiwa False Negative (FN) akan memunculkan dampak yang fatal. Ketika FN beresiko tinggi, maka metrik yang digunakan tidak cukup hanya menggunakan akurasi saja, namun juga Recall. Apabila ada 1 saja file malware yang lolos dan gagal dideteksi, maka komputer korban akan beresiko tinggi, mulai dari komputer menjadi lambat, munculnya celah keamanan data, hingga hilangnya atau terjadinya

Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 5 No. 3 Juli 2023 Hal. 217-223
http://jurnal.unidha.ac.id/index.php/jteksis

E-ISSN : **2655-8238**
P-ISSN : **2964-2132**

pencurian data. Adapun formula perhitungan Recall/Sensitivity/TPF tampak pada persamaan 5.

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

Berdasarkan penjelasan yang telah dikemukakan di atas, maka dua metrik tersebut lah yang akan digunakan dalam penelitian terkait deteksi malware ini.

HASIL DAN PEMBAHASAN

Dalam penelitian ini, dilakukan 2 kali eksperimen dimana eksperimen pertama untuk mencari algoritma terbaik pada dataset yang telah disediakan, sementara eksperimen kedua untuk meningkatkan performa dari algoritma terbaik tersebut dengan Principal Component Analysis. Pada eksperimen pertama, dilakukan perbandingan 5 algoritma klasifikasi, yaitu Random Forest, Adaboost, Neural Network, SVM, dan kNN. Dengan menggunakan 5-Fold Cross Validation, hasil perbandingan tersebut dapat dilihat pada Tabel 2.

Penelitian ini hanya menggunakan dua metrik evaluasi, yaitu Akurasi dan Recall, mengingat deteksi malware membutuhkan metrik yang mengukur False Negatif dengan baik. Dengan demikian, akan bisa dideteksi algoritma mana yang paling jarang menghasilkan False Negatif mengingat resiko besar yang dihadapi apabila terjadi kegagalan dalam mendeteksi file malware karena dianggap sebagai goodware. Oleh karena itu, sangat penting untuk mengukur Recall dibandingkan metrik lain seperti presisi hingga specificity. Sementara itu, metrik akurasi digunakan untuk mengukur keakuratan suatu algoritma mendeteksi dengan benar. Akurasi merupakan metrik yang umum digunakan dalam setiap kasus klasifikasi.

Tabel 2. Hasil perbandingan 5 algoritma klasifikasi

Ranking	Algoritma	Akurasi	Recall
1	Random Forest	0.983	0.983
2	Adaboost	0.982	0.982
3	Neural Network	0.967	0.967
4	SVM	0.958	0.958
5	kNN	0.942	0.942

Berdasarkan data yang diperoleh pada Tabel 2, Random Forest tampak mengungguli 4 algoritma klasifikasi lain, dimana Random Forest memiliki skor Akurasi tertinggi, yaitu 98.3%, serta Recall 98.3% juga. Sementara itu, skor terendah dimiliki oleh kNN, dimana hanya mampu menghasilkan skor akurasi 94.2% juga Recall sebesar 94.2%. Dengan hasil ini, maka Random Forest dipilih sebagai algoritma yang akan ditingkatkan performanya, dengan mengaplikasikan Principal Component Analysis untuk reduksi fiturnya.

Mengingat rekomendasi variance yang digunakan di atas angka 80%, maka 10 data diambil dengan nilai variance berada di rentang 80%-86%. Semakin tinggi variance, semakin tinggi pula jumlah fitur digunakan. Maka dari itu, nilai variance pun dibatasi pada angka 86%. Adapun jumlah Principle Components (PC) yang digunakan berada pada rentang 25 hingga 34. Hasil penerapan PCA pada algoritma Random Forest dapat dilihat pada Tabel 3.

Tabel 3. Hasil penerapan PCA pada algoritma Random Forest

Jumlah PC	Variance	Akurasi	Recall
25	80%	0.9828	0.9828
26	80%	0.9824	0.9824
27	81%	0.9832	0.9832
28	82%	0.9836	0.9836
29	82%	0.9832	0.9832
30	83%	0.9814	0.9814
31	84%	0.9828	0.9828
32	84%	0.9870	0.9870
33	85%	0.9830	0.9830
34	86%	0.9814	0.9814

Pada tabel 3, terlihat bahwa PCA bekerja dengan baik pada algoritma Random Forest dan juga dataset yang telah disiapkan. Sebagai bukti, jumlah PC dalam kisaran 25-34 dapat mewakili nilai variance di atas 80%, dengan hasil akurasi dan recall yang tidak jauh berbeda dengan algoritma Random Forest tanpa PCA. Jumlah fitur yang semula berjumlah 1084 kini telah berhasil diturunkan secara signifikan dengan menggunakan PCA. Bahkan dengan jumlah PC sebanyak 32 variance sebeesar 84%, skor akurasi serta recall mengalami peningkatan dibandingkan tanpa PCA. Apabila skor akurasi dan recall pada algoritma Random Forest tanpa adanya PCA berada di angka 98.3%, maka setelah diterapkannya PCA, skor akurasi dan recall algoritma Random Forest mengalami peningkatan menjadi 98.44%. Sedikit peningkatan ini tentu sangat penting untuk mencapai target zero tollerance pada deteksi malware, khususnya pada metrik Recall.

Pembahasan lebih dalam dapat dilihat dengan membandingkan nilai-nilai yang ada pada confusion matrix. Gambar 2 menunjukkan confusion matrix dari Random Forest tanpa reduksi fitur dengan PCA. Hasilnya, dari 1190 upaya klasifikasi yang dilakukan, algoritma Random Forest gagal mengklasifikasikan 10 file malware sebagai malware. Artinya, dari 595 file malware yang diprediksi, terdapat 10 yang gagal diprediksi dengan benar (False Negative). Pada confusion matrix tersebut, ‘0’ mewakili malware, sementara ‘1’ mewakili goodware. Jika berbicara tentang malware yang notabene False Negative beresiko tinggi, maka satu pun kegagalan dalam mendeteksi atau

Jurnal Teknologi Dan Sistem Informasi Bisnis
Vol. 5 No. 3 Juli 2023 Hal. 217-223
http://jurnal.unidha.ac.id/index.php/jteksis

E-ISSN : **2655-8238**
P-ISSN : **2964-2132**

mengklasifikasi dapat mengakibatkan kerugian pada user, berupa kerusakan, pada sistem ataupun data. Oleh karena itu, upaya untuk meningkatkan skor Recall harus terus dilakukan.

		Prediksi		
		0	1	∑
Aktual	0	585	10	595
	1	10	585	595
	∑	595	595	1190

Gambar 2. Confusion Matrix untuk Random Forest tanpa PCA

		Prediksi		
		0	1	∑
Aktual	0	589	6	595
	1	10	585	595
	∑	599	591	1190

Gambar 3. Confusion Matrix untuk Random Forest dengan PCA

Gambar 3 menunjukkan hasil confusion matrix dari algoritma Random Forest dengan menerapkan PCA untuk reduksi fitur. Hasilnya, terdapat peningkatan deteksi file malware. Di sisi lain, False Negative juga berhasil dikurangi. Dengan PCA, False Negative yang semula berjumlah 10, turun menjadi 6. Penurunan skor False Negative ini dinilai cukup baik, karena semakin sedikit malware yang gagal dideteksi.

Meskipun demikian, pada penelitian ini belum menunjukkan penurunan pada kasus False Positive, meskinpun PCA sudah diimplementasikan. Angka False Positive yang semua 10, masih bertahan di posisi yang sama. Namun pada kasus ini, peneliti fokus pada penurunan angka False Negative daripada False Positive. Keseimbangan penurunan angka False Negative dan False Positive akan dilaksanakan pada penelitian selanjutnya.

SIMPULAN

Penelitian tentang peningkatan performa algoritma Random Forest menggunakan PCA telah selesai dilakukan. Random Forest dipilih karena memiliki performa Akurasi dari Recall terbaik dibandingkan 4 algoritma lain, seperti: Adaboost, Neural Network, Support Vector Machine dan k-Nearest Neighbor. Performa yang diperoleh Random Forest pada Dataset malware yang telah disiapkan adalah 98.3%, baik untuk Akurasi maupun Recall. Selanjutnya eksperimen dilakukan dengan melakukan fitur reduksi pada dataset, mengingat dataset yang digunakan tersebut memiliki fitur berjumlah 1084. Setelah fitur direduksi menjadi 32 PC dan memperhatikan jumlah Variance pada angka 84%, performa algoritma Random Forest meningkat menjadi 98.44%. Berdasarkan data pada tabel confusion matrix, dari total 1190 file dimana 595 diantaranya adalah file malware, algoritma Random Forest yang semula memiliki False Negative berjumlah 10, berhasil dikurangi menjadi 6 setelah dilakukannya reduksi fitur dengan PCA. Hal ini menunjukkan adanya peningkatan pada algoritma Random Forest, karena semakin sedikit jumlah kesalahan deteksi, khususnya mendeteksi file

Pada penelitian selanjutnya, penurunan angka false negative dan false positive perlu diperhatikan untuk mencapai prinsip zero tollerance dalam deteksi malware. Selain itu, dataset malware juga perlu diupdate, mengingat dataset yang digunakan ini dibuat pada tahun 2019 sementara perkembangan malware terus meningkat dengan pesat.

UCAPAN TERIMAKASIH

Terimakasih ditujukan kepada Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LPPM) dan Fakultas Ilmu Komputer, Universitas Dian Nuswantoro atas dukungan fasilitas dan pendanaan pada penelitian ini.

DAFTAR PUSTAKA

[1] O. Aslan and R. Samet, "A Comprehensive Review on Malware Detection Approaches," *IEEE Access*, vol. 8, pp. 6249–6271, 2020, doi: 10.1109/ACCESS.2019.2963724.
[2] F. A. Rafrastara, C. Supriyanto, C. Paramita, and Y. P. Astuti, "Deteksi Malware menggunakan Metode Stacking berbasis Ensemble," *JPIT*, vol. 8, no. 1, pp. 11–16, 2023.
[3] F. A. Rafrastara, C. Supriyanto, C. Paramita, Y. P. Astuti, and F. Ahmed, "Performance Improvement of Random Forest Algorithm for Malware Detection on Imbalanced Dataset using Random Under-Sampling Method," *JPIT*, vol. 8, no. 2, pp. 113–118, 2023.
[4] N. Shahid *et al.*, "Mathematical analysis and numerical investigation of advection-reaction-diffusion computer virus model," *Results in Physics*, vol. 26, p. 104294, Jul. 2021, doi: 10.1016/j.rinp.2021.104294.
[5] F. A. Rafrastara and F. M. A., "Advanced Virus Monitoring and Analysis System," *IJCSIS*, vol. 9, no. 1, 2011.
[6] F. A. Rafrastara, *Belajar Membuat Virus Komputer Mulai dari NOL*. Semarang, Indonesia: NeomediaPress, 2007.
[7] H. Shah and D. M. G. Comissiong, "Computer Virus Model with Stealth Viruses and Antivirus Renewal in a Network with Fast Infectors," *SN Computer Science*, vol. 2, no. 5, pp. 1–8, 2021, doi: 10.1007/s42979-021-00780-9.
[8] A. Pratama and F. A. Rafrastara, "Computer Worm Classification," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 10, no. 4, pp. 21–24, 2012.
[9] A. Nugraha and F. A. Rafrastara, "TAXONOMY BOTNET DAN STUDI KASUS: CONFICKER," in *Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2011 (Semantik 2011)*, Semarang, Indonesia, 2011.
[10] A. Nugraha and F. A. Rafrastara, "BOTNET DETECTION SURVEY," in *Seminar Nasional Teknologi Informasi &*

E-ISSN : **2655-8238**
P-ISSN : **2964-2132**

Komunikasi Terapan 2011 (Semantik 2011), Semarang, Indonesia, 2011.

[11] A. Adriyendi and Y. Melia, "KLASIFIKASI MENGGUNAKAN NAÏVE BAYES DAN K-NEAREST NEIGHBOR PADA MANAJEMEN LAYANAN TEKNOLOGI INFORMASI," *JTEKSIS*, vol. 2, no. 2, pp. 99–107, Jul. 2020, doi: 10.47233/jteksis.v2i2.121.

[12] M. Afdhal, V. Ariandi, and R. Rita, "Memprediksi Penjualan Pada Toko Hanifah Metode C.45," *JTEKSIS*, vol. 4, no. 2, pp. 248–255, Jul. 2022, doi: 10.47233/jteksis.v4i1.460.

[13] F. Hidayat and T. M. S. Astsauri, "Applied random forest for parameter sensitivity of low salinity water Injection (LSWI) implementation on carbonate reservoir," *Alexandria Engineering Journal*, vol. 61, no. 3, pp. 2408–2417, 2022, doi: 10.1016/j.aej.2021.06.096.

[14] F. C. C. Garcia and F. P. Muga, "Random Forest for Malware Classification," pp. 1–4, 2016.

[15] H. J. Zhu, T. H. Jiang, B. Ma, Z. H. You, W. L. Shi, and L. Cheng, "HEMD: a highly efficient random forest-based malware detection framework for Android," *Neural Computing and Applications*, vol. 30, no. 11, pp. 3353–3361, 2018, doi: 10.1007/s00521-017-2914-y.

[16] B. M. Khammas, "Ransomware Detection using Random Forest Technique," *ICT Express*, vol. 6, no. 4, pp. 325–331, 2020, doi: 10.1016/j.icte.2020.11.001.

[17] A. Mishra, A. M. K. Cheng, and Y. Zhang, "Intrusion Detection Using Principal Component Analysis and Support Vector Machines," in *2020 IEEE 16th International Conference on Control & Automation (ICCA)*, Singapore: IEEE, Oct. 2020, pp. 907–912. doi: 10.1109/ICCA51439.2020.9264568.

[18] G. H. M., T. B. Adji, and N. A. Setiawan, "Penggunaan Metodologi Analisa Komponen Utama (PCA) untuk Mereduksi Faktor-Faktor yang Mempengaruhi Penyakit Jantung Koroner," in *Proceeding Seminar Nasional SciETec (Science, Engineering and Technology) 2012*, Malang: Program Magister dan Doktor, Fakultas Teknik, Universitas Brawijaya, Feb. 2012.

[19] E. Kartikadarma, S. Wijayanti, S. A. Wulandari, and F. A. Rafrastara, "Principle Component Analysis for Classification of the Quality of Aromatic Rice," *IJCSIS*, vol. 15, no. 8, pp. 315–319, 2017.

[20] R. A. Johnson and D. W. Wichern, *Applied multivariate statistical analysis*, 5th ed. Upper Saddle River, N.J: Prentice Hall, 2002.

[21] E. Kaloyanova, "What Is Principal Components Analysis?," *365 DataScience*, Oct. 20, 2021. https://365datascience.com/tutorials/python-tutorials/principal-components-analysis/ (accessed May 29, 2023).