DETEKSI ANOMALY PDF MALWARE PADA AGREGATOR NASIONAL (GARUDA) KEMDIKBUD DIKTI DENGAN SUPPORT VECTOR MACHINE

OLEH:

RANI OCTAVIANI

09011281823047

JURUSAN SISTEM KOMPUTER

FAKULTAS ILMU KOMPUTER

UNIVERSITAS SRIWIJAYA

2022

LEMBAR PENGESAHAN

DETEKSI ANOMALY PDF MALWARE PADA AGREGATOR NASIONAL (GARUDA) KEMDIKBUD DIKTI DENGAN SUPPORT VECTOR MACHINE

TUGAS AKHIR

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

Oleh
RANI OCTAVIANI
09011281823047

Indralaya, Agustus 2022
Mengetahui,

Pembimbing Tugas Akhir I	Pembimbing Tugas Akhir II
Deris Stiawan, M.T., Ph.D	Tri Wanda Septian. S.Kom., M.Sc
NIP 197806172006041002	NIK.1901062809890001

Ketua Jurusan Sistem Komputer

Dr. Ir. Sukemi, M.T.
NIP.196612032006041001

HALAMAN PERSETUJUAN

Telah diuji dan lulus pada:

Hari : Kamis

Tanggal : 14 Juli 2022

Tim Penguji :

1. Ketua : **Ahmad Zarkasi, S.T., M.T.** 9/8/2022.

2. Sekretaris : **Adi Hermansyah, M.T.**

3. Penguji : **Ahmad Heryanto, M.T.**

4. Pendamping I : **Deris Stiawan, M.T., Ph.D**

5. Pendamping ii : **Tri Wanda Septian, M.Sc**

Mengetahui,

Ketua Jurusan Sistem Komputer

Dr. Ir. H. Sukemi M.T.

NIP. 196612032006041001

HALAMAN PERNYATAAN

Yang bertanda tangan dibawah ini:

Nama : Rani Octaviani
NIM : 09011281823047
Judul : Deteksi Anomaly PDF Malware Pada Agregator (Garuda) Kemdikbud Dikti dengan Support Vector Machine

Hasil Pengecekan Software *iThenticate/Turnitin* : 5%

Menyatakan bahwa laporan tugas akhir saya merupakan hasil karya sendiri dan bukan hasil penjiplakan atau plagiat. Apabila ditemukan unsur penjiplakan atau plagiat dalam laporan tugas akhir ini, maka saya bersedia menerima sanksi akademik dari universitas Sriwijaya.

Demikian, pernyataan ini saya buat dalam keadaan sadar dan tidak dipaksakan.

Indralaya, Juli 2022

Rani Octaviani
09011281823047

KATA PENGANTAR

Puji syukur kepada Allah SWT atas rahmat-Nya sehingga penulis diberi kesempatan untuk menyelesaikan Tugas Akhir yang berjudul **"Deteksi Anomaly PDF Malware pada Agregator Nasional (Garuda) Kemdikbud Dikti dengan Support Vector Machine".** Pada kesempatan ini, penulis menyampaikan rasa terima kasih kepada semua pihak yang telah membantu dan mendukung sehingga dapat memberikan dorongan kepada penulis dalam penyelesaian Tugas Akhir ini.

Oleh karena itu, penulis mengucapkan rasa terima kasih kepada:

- Ayah dan Ibu yang terus memberikan do'a restu dan dukungan selama menempuh perkuliahan serta kedua adik saya yang selalu memberikan energi positif kepada saya,
- Bapak Deris Stiawan, M.T., Ph.D selaku Dosen Pembimbing Akademik sekaligus Pembimbing I Tugas Akhir,
- Kak Tri Wanda Septian, S.Kom.,M.Sc dan Mbak Nurul Afifah M.Kom yang selalu membimbing serta memberi masukan selama penelitian,
- Kak Meutia Zamieyus yang selalu memberikan saran serta masukan selama pengerjaan tugas akhir,
- Teman – teman sekalian; Thesa, Nana, Gavira, Furqon, Arif dan teman lainnya yang selalu berbagi keluh kesah dan suka cita,
- Teman – teman di Prabumulih; Ferjielia, Indah, Muthiah, dan teman lainnya yang selalu memberikan dukungan,

- Seluruh dosen, staff, serta karyawan Fakultas Ilmu Komputer Universitas Sriwijaya,

- Almamater.

Penulis menyadari bahwa dalam penulisan Tugas Akhir ini masih banyak terdapat kekurangan, oleh karena itu seluruh saran dan kritik sangatlah berguna untuk menjadi bahan evaluasi bagi penulis.

Indralaya 2022
Penulis

Rani Octaviani
NIM.09011281823047

ANOMALY DETECTION OF PDF MALWARE ON NATIONAL AGREGATOR (GARUDA) KEMDIKBUD DIKTI USING SUPPORT VECTOR MACHINE

RANI OCTAVIANI (09011281823047)

Computer Engineering Department, Computer Science Faculty, Sriwijaya University

Email: octavianirani0@gmail.com

ABSTRACT

Garba Rujukan Digital (GARUDA) is a repository for articles in PDF files. All PDF files have metadata that can be used by hackers so that PDF files become PDF Malware. In the metadata of PDF Malware there are encrypting code that is used by hackers to run malware. Therefore, the focus of this research is the detection of anomalies from PDF Malware based on metadata obtained from the extraction using PDFiD, then it is used as a feature for the multiclass classification process, namely PDF Benign, PDF-html, and PDF-Malware using Support Vector Machines. The results of the classification with the RBF kernel obtained a precision value of 88.67%, recall of 87%, F-1 Score of 86.67% and 87% accuracy and Support Vector Machine with Polynomial kernel obtained a precision value of 83%, recall of 80%, F1 -Score of 78.3% and accuracy of 80%.

Keywords: *PDF Malware, PDFiD, Multiclass, Support Vector Machine.*

Acknowledged By,

Supervisor I	Supervisor II
Deris Stiawan, M.T., Ph.D	**Tri Wanda Septian. S.Kom., M.Sc**
NIP. 197806172006041002	**NIK. 1901062809890001**

Head of Computer System Department

3/8/22

Dr. Ir. Sukemi, M.T.

NIP.196612032006041001

Deteksi Anomaly PDF Malware Pada Agregator Nasional (Garuda) Kemdikbud Dikti dengan *Support Vector Machine*

RANI OCTAVIANI (09011281823047)

Jurusan Sistem Komputer, Fakultas Ilmu Komputer, Universitas Sriwijaya
Email: octavianirani0@gmail.com

ABSTRAK

Garba Rujukan Digital (GARUDA) merupakan sebuah repositori penyimpanan artikel – artikel dalam bentuk PDF, dimana setiap PDF memiliki metadata yang dapat dimanfaatkan oleh *hacker* untuk melakukan tindak kejahatan yang menyebabkan adanya PDF Malware. Pada metadata dari PDF Malware terdapat *encrypting code* yang digunakan oleh *hacker* untuk menjalankan malware, oleh karena itu, fokus penelitian ini adalah deteksi anomaly dari PDF Malware berdasarkan metadata yang diperoleh dari hasil ekstraksi menggunakan PDFiD dari setiap PDF, yang kemudian digunakan sebagai fitur untuk proses klasifikasi *multiclass* yaitu PDF Benign, PDF-html, dan PDF-Malware menggunakan *Support Vector Machine*. Hasil klasifikasi dengan kernel RBF memperoleh nilai presisi sebesar 88.67%, recall sebesar 87%, F-1 Score sebesar 86,67% dan akurasi 87% dan Support Vector Machine dengan kernel Polynomial memperoleh nilai presisi sebesar 83%, recall sebesar 80%, F1-Score sebesar 78,3% dan akurasi sebesar 80%.
Kata Kunci: PDF Malware, PDFiD, *Multiclass*, *Support Vector Machine*.

Mengetahui

Pembimbing Tugas Akhir I	Pembimbing Tugas Akhir II
Deris Stiawan, M.T., Ph.D	**Tri Wanda Septian, S.Kom., M.Sc**
NIP. 197806172006041002	**NIK. 1901062809890001**

Ketua Jurusan Sistem Komputer

Dr. Ir. Sukemi, M.T.

NIP.196612032006041001

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1. Latar Belakang

Garba Rujukan Digital (GARUDA) merupakan sebuah repositori penyimpanan yang menjadi sumber kumpulan informasi yang mencakup beberapa aspek seperti ilmu pengetahuan, ilmu perilaku, ilmu moral, matematika dan komputer dan lainnya yang berhubungan dengan publikasi ilmiah yang dikelola oleh KEMDIKBUD DIKTI. Artikel yang terdapat pada portal repositori ini dipublikasikan dalam bentuk PDF pada sebuah cloud penyimpanan yang mana bisa saja terdapat celah sehingga dapat dimanfaatkan oleh *hacker* untuk melakukan tindak kejahatan misalnya pengarahan pengunduhan dokumen atau aplikasi bahaya, serangan yang telah ditargetkan dan pengiriman e-mail secara acak. Celah dari file PDF yang dimanfaatkan oleh *hacker* untuk memudahkan dalam memenuhi tujuannya menyebabkan adanya PDF Malware yang mana berisikan kode mencurigakan yang tertanam pada file [1] atau kode yang akan diinterpertasi atau dieksekusi oleh perangkat lunak untuk membaca file [2],[3] Pada file dengan format PDF, terdapat beberapa fitur yang salah satunya menjadi tempat dimana kode mencurigakan ditanamkan agar dapat diinterpretasikan oleh prangkat pembaca file sesuai dengan tujuan dari *hacker* [4].

Machine Learning merupakan algoritma visi komputer yang mana mampu melakukan proses klasifikasi dan identifikasi data seperti gambar, video, teks dan lainnya secara otomatis [5]. *Machine Learning* digunakan untuk mendeteksi *malicious PDF files* pada penelitian sebelumnya [6],[7] salah satunya adalah penggunaan algoritma *Support Vector Machine* (SVM). Algoritma *Support Vector Machine* adalah teknik *supervised learning* yang menentukan *hyperplanes* terbaik agar dapat memisahkan kelas yang telah diberikan.

Pada [8] algortima *Support Vector Machine* mendapatkan nilai *true positive* yang lebih baik dibandingkan dengan algoritma *decision tree*, penelitian [9] dimana algoritma TDIF dikombinasikan dengan beberapa

metode yaitu SVM, KNN, dan DT dan dari penelitian ini kombinasi dari TDIF dan SVM memperoleh nilai akurasi terbaik diantara dua metode lainnya yaitu sebesar 99.08%, penelitian lainnya [10] menunjukkan bahwa algoritma SVM dapat melakukan proses deteksi dengan baik serta memperoleh nilai akurasi yang cukup baik.

Pada tugas akhir ini, penulis akan melakukan deteksi dari malicious PDF files berdasarkan fitur yang terdapat pada file pdf. Fitur tersebut diperoleh dari hasil ekstrak file pdf menggunakan pdfid.py dan digunakan sebagai dataset untuk masuk ke machine learning agar dapat dideteksi dan menjadi bahan analisa yang dapat digunakan sebagai referensi. Adapun judul dari tugas akhir ini adalah "Deteksi Anomaly PDF Malware Pada Agregator Nasional (GARUDA) Kemdikbud Dikti dengan *Support Vector Machine*".

1.2. Rumusan Masalah

Berikut ini merupakan rumusan masalah dari penelitian Tugas Akhir yang dilakukan:

1. Bagaimana proses persiapan data untuk deteksi PDF Malware?
2. Bagaimana cara validasi terhadap *imbalance data* untuk pengambilan sampel *training* dan *testing*?
3. Bagaimana cara perlakuan *imbalance data* agar mendapat performa terbaik?
4. Bagaimana pengaruh dari RBF dan Polynomial kernel dari metode *Support Vector Machine* terhadap nilai presisi, recall, akurasi dan f-1 score?

1.3. Batasan Masalah

Berikut ini merupakan batasan masalah dari penelitian Tugas Akhir yang dilakukan:

1. Dataset yang digunakan untuk penelitian tugas akhir adalah pdf yang berasal dari repositori GARUDA,
2. *Support Vector Machine* merupakan metode yang digunakan untuk mengklasifikasikan Anomali PDF Malware,
3. Hanya menggunakan kernel RBF dan Polynomial dari *Support Vector Machine.*

1.4. Tujuan

Berikut ini merupakan tujuan dari penelitian Tugas Akhir yang dilakukan:

1. Menyediakan dataset untuk proses deteksi PDF Malware,
2. Melakukan pengambilan data sampel untuk *training* dan *testing*,
3. Menerapkan SMOTE pada dataset yang digunakan untuk proses deteksi PDF Malware,
4. Menganalisa hasil kinerja dari proses klasifikasi dengan kernel yang digunakan pada metode *Support Vector Machine.*

1.5. Manfaat

Berikut ini merupakan manfaat dari penelitian Tugas Akhir yang dilakukan:

1. Dataset dari PDF GARUDA Repository dapat digunakan untuk deteksi PDF Malware,
2. Stratified KFold mampu memberikan sampel data untuk deteksi PDF Malware,
3. SMOTE dapat membuat dataset baru dari kelas minoritas untuk mengatasi *imbalance dataset,*
4. Validasi hasil dari kedua kernel yang digunakan pada metode *Support Vector Machine* sebagai metode klasifikasi.

1.6. Metodologi Penelitian

Terdapat tahapan – tahapan yang dilakukan dalam penelitian tugas akhir, yaitu:

1. Tahapan Pertama (Studi Pustaka)

   Aktivitas yang dilakukan pada tahapan ini adalah mengumpulkan informasi yang berkaitan dengan topik penelitian tugas akhir yang dilakukan dari sumber – sumber yang akurat seperti jurnal ilmiah, buku, artikel serta sumber lainnya.
2. Tahapan Kedua (Perancangan Sistem)

   Aktivitas yang dilakukan pada tahapan ini adalah merancang sistem agar dapat melakukan klasifikasi dari anomali PDF Malware dengan menggunakan algoritma *Support Vector Machine* pada bahasa pemrograman python.

3. Tahapan Ketiga (Pengujian dan Pengambilan Data)

   Aktivitas yang dilakukan pada tahapan ini adalah mengumpulkan pdf dari repositori GARUDA untuk dilakukan *scanning* dan ekstraksi fitur data sebagai csv.
4. Tahapan Keempat (Hasil dan Analisa)

   Langkah yang dilakukan selanjutnya adalah melakukan pengolahan data seperti memberikan label, *oversampling data*, menerapkan *cross validation*, setelah itu menerapkan algoritma yang digunakan untuk dapat dianalisa.
5. Tahapan Kelima (Kesimpulan dan Saran)

   Langkah akhir yang dilakukan adalah menarik kesimpulan dari penelitian yang telah dilakukan dan memberikan saran agar dapat dikembangkan untuk penelitian selanjutnya.

1.7. Sistematika Penulisan

Berikut ini merupakan sistematika yang digunakan dalam penulisan tugas akhir agar mendeskripsikan bab – bab yang terdapat dalam tugas akhir yang dilakukan:

BAB I. PENDAHULUAN

Isi yang terdapat pada BAB I adalah latar belakang, tujuan penelitian, manfaat penelitian, rumusan masalah, batasan masalah, metodologi penelitian, dan sistematika penulisan.

BAB II. TINJAUAN PUSTAKA

Isi yang terdapat pada BAB II adalah menjelaskan tentang PDF Malware, dataset yang digunakan, oversampling, cross validation, algoritma yang digunakan dan validasi dari penelitian yang dilakukan.

BAB III. METODOLOGI

Isi yang terdapat pada BAB III adalah mengenai tahapan yang dilakukan dalam penelitian tugas akhir serta rancangan sistem yang digunakan untuk sistem klasifikasi yang diterapkan pada penelitian tugas akhir.

BAB IV. HASIL DAN ANALISIS

Isi yang terdapat pada BAB IV adalah hasil klasifikasi dari algoritma yang digunakan serta analisa terhadap sebab keadaan dari hasil yang diperoleh.

BAB V. KESIMPULAN DAN SARAN

Isi yang terdapat pada BAB V adalah memberikan kesimpulan berdasarkan hasil yang telah diperoleh, kemudian memberikan saran agar dapat dilakukan pengembangan untuk penelitian selanjutnya

BAB II
TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Pada penelitian yang dilakukan, terdapat beberapa rujukan mengenai penelitian terkait yang telah dilakukan sebelumnya yang mana digunakan dalam melakukan penelitian tugas akhir ini. Pada penelitian terkait yang menjadi rujukan penelitian, terdapat beberapa algoritma *machine learning* yang digunakan seperti yang terlampir pada Tabel 2.1.

Tabel 2. 1 Daftar Penelitian Terkait

No	Nama Penulis	Judul Penelitian	Dataset	Metode
1	Pavel Laskov, Nedim Srndic [2]	Static Detection of Malicious JavaScript-Bearing PDF Documents	VirusTotal	One-Class SVM
2	Pavel Laskov, Nedim Srndic [8]	Detection of Malicious PDF Files Based on Hierarchical Document Structure	VirusTotal, Google	Decision Trees, SVM
3	Abdachul Charim, Setio Basuki, Denar Regata Akbi [11]	Detect Malware in Portable Document Format Files (PDF) Using Support Vector Machine and Random Forest	*Various Sources*	SVM, Random Forest
4	Pavel Laskov, Nedim Srndic [12]	Hidost: A Static Machine-Learning-Based Detector of Malicious Files	VirusTotal	SVM, Random Forest
5	Davide Maiorca, Giorgio Giacinto	Structural and Content-Based	Contagio	Decision Trees

	[13]	Approach for a Precise and Robust Detection Malicious PDF Files		
6	Davide Maiorca, Giorgio Giacinto, Igino Corona [14]	A Pattern Recognition System for Malicious PDF Files Detection	Contagio, Yahoo	Naive Bayes, SVM, Decision Trees
7	Bonan Cuan, Alienor Damien, Claire Delaplace, Mathieu Valois [7]	Malware Detection in PDF Files Using Machine Learning	Contagio	SVM
8	Md Mursalin, Yuan Zhang, Yuehui Chen, Nitesh V Chawla [15]	Automated Epileptic Seizure Detection Using Improved Correlation-based Feature Selection with Random Forest Classifier	Department of Epileptology, Universisty of Bonn	Random Forest
9	Baigal tugsSanjaa, Erdenebat Chuluun [16]	Malware Detection Using Linear SVM	VX Heaven	Linear SVM
10	Charles Smutz, Angelos Stavrou [17]	Malicious PDF Detection using Metadata and Structural Features	Contagio	Random Forest

Penelitian tugas akhir ini merujuk ke penelitian [11] yang menggunakan dua metode *machine learning* yaitu *Support Vector Machine* dan *Random*

Forest untuk melakukan klasifikasi data *PDF Malware.* Tabel 2.2 menunjukkan perbandingan penelitian yang dilakukan dengan penelitian sebelumnya yang menjadi rujukan dari penelitian tugas akhir.

Tabel 2. 2 Perbandingan Penelitian dengan Penelitian Sebelumnya

No	**Penjelasan**	**Penelitian Terkait** [11]	**Penelitian yang Diajukan**
1	Algoritma	Random Forest dan SVM	SVM
2	Jenis Klasifikasi	Binary	Multiclass

2.2. PDF Malware

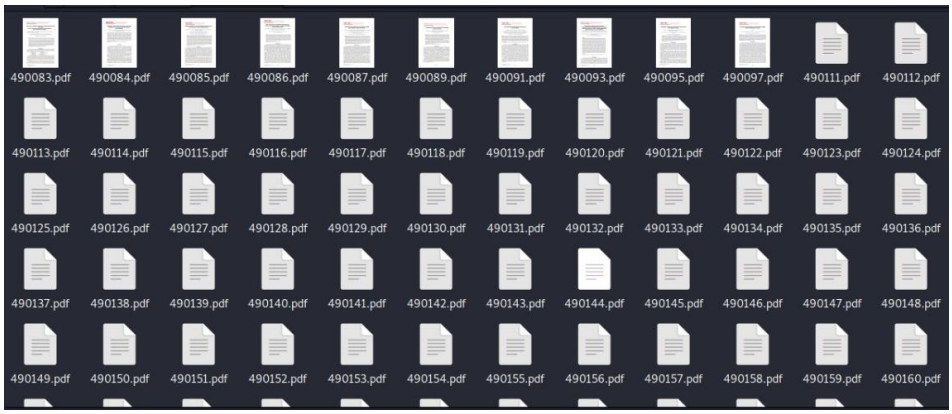
Sebuah file PDF merupakan file yang mudah untuk dilakukan pengeditan dan dimanipulasi dikarenakan berisikan teks, dan memberikan sedikit batasan kepada *hacker*. Sejak tahun 2001, Adobe memberikan penambahan format yaitu tambahan kompresi, algoritma enkripsi, *scripting support*, dan *multimedia support* [4]. Berdasarkan penambahan format yang diberikan oleh Adobe memberikan kemudahan atau celah yang dimanfaatkan untuk menanamkan *malware* ke dalam sebuah file PDF atau menyembunyikan keberadaan dari *malware* yang telah ditanamkan ke sebuah header file PDF [18].

Pada file DF terdapat *header* yang berisikan fitur – fitur [19] [20] yang dapat dijadikan sebagai karakterisik yang berguna dalam membantu klasifikasi PDF *Malware* dan PDF normal [17] ketika dataset masuk dan diolah ke dalam *machine learning* menggunakan algoritma yang dipilih yaitu *support vector machine.*

2.3. Dataset PDF Malware

Dataset yang digunakan dalam pengerjaan tugas akhir ini adalah berasal dari kumpulan PDF dari repositori Garba Rujukan Digital (GARUDA) yang memiliki ekstensi format pdf seperti pada Gambar 2.1. Dataset yang digunakan

berisikan PDF normal dan PDF *malware* yang diketahui setelah dilakukan pengecekan secara satu persatu menggunakan *online checker* yaitu VirusTotal.



Gambar 2. 1 Dataset PDF GARUDA

2.4. Ekstraksi Dataset

Ekstraksi yang digunakan dalam pengerjaan tugas akhir ini adalah melakukan *parsing* terhadap tiap file PDF pada dataset agar didapatkan *header* yang berguna sebagai fitur untuk dataset [6].

2.4.1. PDFID

Dalam melakukan ekstraksi fitur dataset untuk pengerjaan tugas akhir ini, *tools* yang dirancang oleh Didier Stevens digunakan yaitu PDFiD. PDFiD merupakan *script python* yang dapat melakukan analisa secara statis terhadap sebuah file PDF [19]. *Script* yang telah dirancang akan melakukan penijauan terhadap sebuah file PDF, dan melakukan perhitungan terhadap nilai yang terdapat pada setiap fitur yang ada. Terdapat sebanyak dua puluh satu fitur yang umumnya ditemukan pada sebuah file yang mencurigakan. Setelah nilai dari fitur didapatkan, nilai dan fitur dijadikan dataset ke dalam bentuk .csv [11].

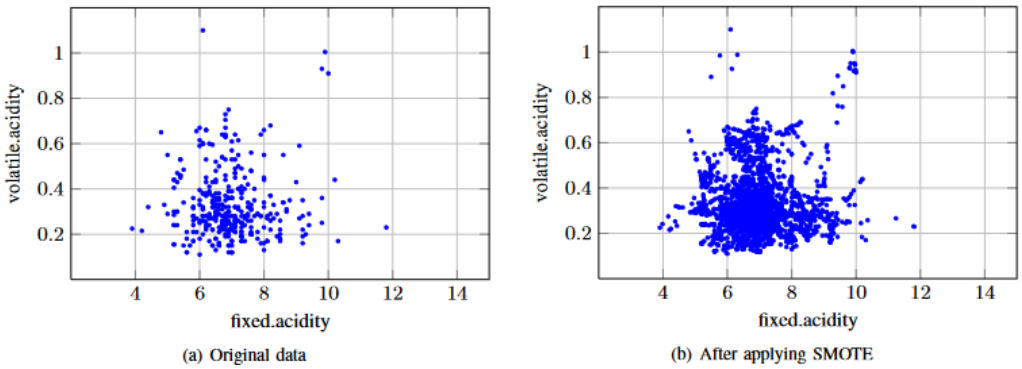
2.5. SMOTE

Performa dari algoritma *machine learning* biasanya dievaluasi menggunakakan akutasi prediksi namun, pada saat data yang digunakan merupakan *imbalanced dataset* maka akan memberikan hasil yang tidak baik karena data yang minor akan tertutupi oleh data yang mayoritas [21]. Oleh

karena itu, perlu dilakukan *oversampling* terhadap dataset yang digunakan. Salah satu teknik untuk *oversampling* dataset adalah *Synthetic Minority Over-Sampling Technique* (SMOTE).

SMOTE akan menghasilkan sample minoritas sintetis untuk mengambil kelas minoritas sehingga diperoleh sample minoritas baru. Cara kerja dari metode ini adalah mencari sample terdekat dengan sample minor yang dipilih secara acak sesuai dengan tingkat sampling yang diberikan, lalu diperoleh sample sintetis baru yang dihasilkan disepanjang garis antara sample minoritas dan tetangga terdekat yang dipilih [22].

Pada penelitian [23] dengan menggunakan algortima klasifikasi *Decission Tree* dan *Random Forest* yang dikombinasikan dengan SMOTE diperoleh nilai akurasi sebesar 92.8% untuk kombinasi *Decision Tree* dan SMOTE sedangkan untuk kombinasi *Random Forest* dan SMOTE diperoleh nilai akurasi sebesar 94.6%. Gambar 2.2 menunjukkan bahwa SMOTE telah berhasil membuat sample baru untuk kelas *minority*.



Gambar 2. 2 Hasil Pengaplikasian SMOTE

2.6. Stratified K-Fold

Metode validasi K-Fold merupakan teknik validasi yang sering kali digunakan yang mana melibatkan pemisahan data *training* dan *testing* dalam bentuk K-Fold. Namun, untuk data penanganan validasi untuk *imballanced dataset* diperlukan tambahan dari teknik validasi tersebut yaitu *Stratified K-Fold* [24], [25]. Teknik validasi ini dapat digunakan untuk masalah klasifikasi dan akan menerapkan distribusi kelas di setiap pemisahan data sehingga sesuai dengan distribusi dalam keseluruhan data *training* [26], [27].

Persamaan yang digunakan untuk menghitung data *training* dalam *Stratified K-Fold* adalah sebagai berikut:

$$\frac{n-1}{n} \times 100\% \tag{2.1}$$

2.7. Support Vector Machine (SVM)

Pertama kali dikenalkan pada tahun 1992 oleh Boser, Guyon, Vapnik, di Annual Workshop yaitu Computional Leraning Theory sebagai konsep pada bidang pengenalan pola [28]. *Support Vector Machine* merupakan algoritma yang digunakan untuk melakukan proses klasifikasi data yang melibatkan dataset training dan testing [16], [29].

Pada umumnya teknik SVM digunakan dalam pemecahan masalah klasifikasi, namun pada beberapa kasus metode ini juga digunakan untuk masalah regresi. Prinsip kerja yang digunakan oleh SVM adalah melakukan prediksi klasifikasi berdasarkan fitur – fitur yang terdapat pada dataset yang digunakan [14], [16]. Algoritma dari SVM akan berusaha mencari *hyperplane* terbaik dengan menggunakan kernel dan hyperparameter yang sesuai dengan ketentuan. Terdapat beberapa parameter yang digunakan dalam algoritma *Support Vector Machine* seperti yang dijelaskan sebagai berikut:

1. Kernel

   Pemilihan kernel yang tepat, berperan penting dalam menentukan *Feature Space* untuk menemukan fungsi *classifier*. Terdapat beberapa kernel pada algoritma ini yaitu: Linear, Polynomial, Radial Basis Function dan Sigmoid. Kernel yang digunakan pada penelitian ini adalah kernel RBF yang dapat menentukan nilai serta lokasi dari *center* dan nilai pembobot secara otomatis dan mampu memiliki rentang yang tak terhingga dan Kernel Polynomial.

2. Nilai C

   Parameter ini berguna untuk mengontrol *trade off* yang terjadi antara *margin* dan error klasifikasi. Penentuan parameter dapat dilakukan dengan beberapa cara, salah satunya ialah *cross validation*. Nilai C yang besar akan memberikan keluaran yang lebih besar terhadap nilai error klasifikasi.

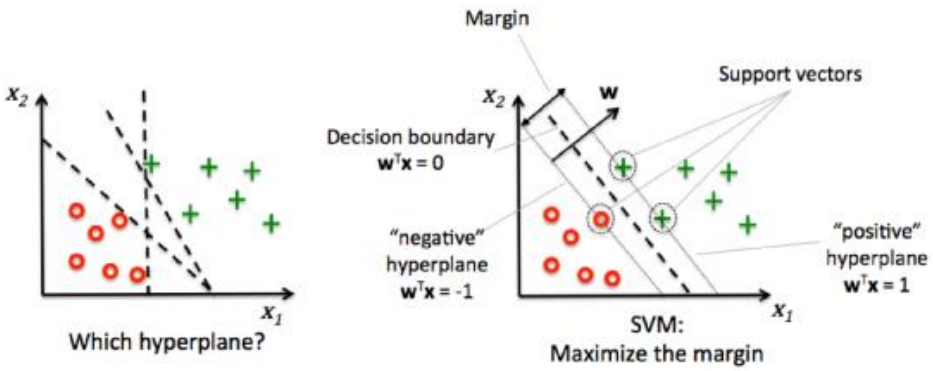
3. Gamma

   Nilai dari parameter Gamma berisikan koefesien dari beberapa kernel yaitu rbf, poly, dan sigmoid dan dapat pula berisikan 1/n_features secara otomatis apabila nilai gamma tidak diberikan.

4. Degree

   Nilai yang terdapat pada parameter Degree memberikan control terhadap fleksibilitas dari *decision boundary*, semakin tinggi nilai yang diberikan maka semakin fleksibel *decision boundary* [30].

Kernel yang biasanya digunakan pada suatu model adalah linear, polynomial, RBF, dan sigmoid.



Gambar 2. 3 Support Vector Machine

Gambar 2.3 menunjukkan poin – poin yang dilalui oleh π+ dan π- disebut *Support Vector*. Persamaan yang digunakan pada Gambar 2.3 dengan margin adalah $= \frac{2}{||\omega||}$ ditunjukkan oleh persamaan – persamaan sebagai berikut:

Persamaan untuk pemisahan *hyperplane*:

$$\pi : \omega^T . x + b = 0 \tag{2.2}$$

Persamaan untuk *positive hyperplane*:

$$\pi^+ : \omega^T . x + b = 1 \tag{2.3}$$

Persamaan untuk *negative hyperplane*:

$$\pi^- : \omega^T . x + b = -1 \tag{2.4}$$

Persamaan 2.5 menunjukkan persamaan yang digunakan oleh kernel *Radial Basis Functions* (RBF).

$$K\ (a,b) = exp(-(a-b)2/2\sigma^2) \quad (2.5)$$

Persamaan 2.6 menunjukkan persamaan yang digunakan oleh kernel Polynomial.

$$K\ (a,b) = (a.b+1)^P \quad (2.6)$$

2.8. Confussion Matrix

Tahap ini dilakukan agar hasil dari rancangan sistem yang telah dimplementasikan merupakan hasil yang valid dan sesuai dengan prediksi yang diharapkan. Dengan menggunakan *confussion matrix* akan diperoleh hasil dari performa klasifikasi berupa output dari dua kelas atau lebih (*multiclass*) yaitu nilai presisi, recall, akurasi dan f-1 score. Gambar 2.4 menunjukkan *confussion matrix* untuk validasi hasil percobaan penelitian tugas akhir. Dalam melakukan validasi, terdapat beberapa persamaan yang digunakan oleh *confussion matrix* [11] seperti yang ditunjukkan sebagai berikut:

- Presisi

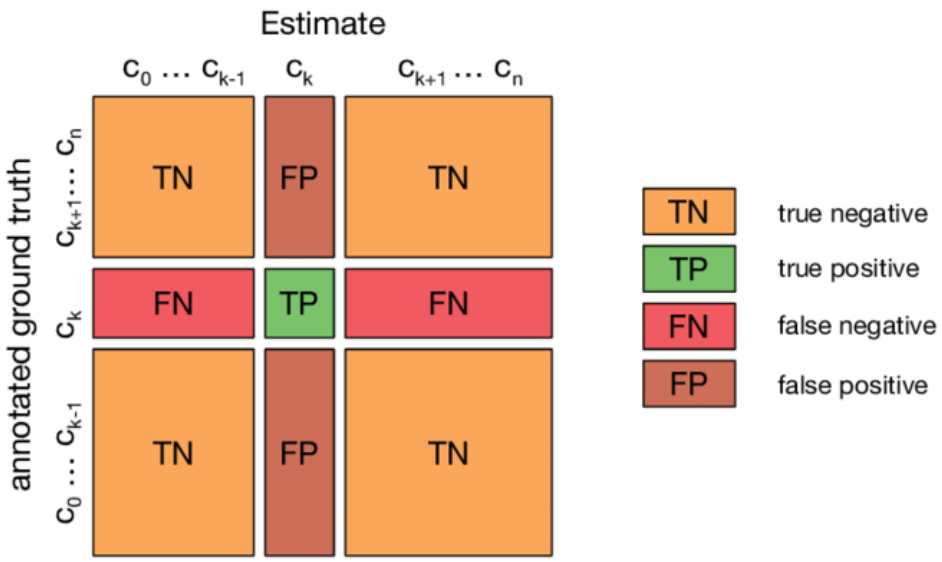
$$\frac{\sum_{i=1}^{l} TP_i}{\sum_{i=1}^{l}(TP_i + FN_i)} \times 100\% \quad (2.6)$$

- Recall

$$\frac{\sum_{i=1}^{l} TP_i}{\sum_{i=1}^{l}(FP_i + TP_i)} \times 100\% \quad (2.7)$$

- Akurasi

$$\frac{\sum_{i=1}^{l} \frac{TP + TN}{TP_i + TN_i + FP_i + FN_i}}{l} \times 100\% \quad (2.8)$$



Gambar 2. 4 Confussion Matrix 3 Kelas

Berikut ini penjelasan dari persamaan di atas:

True Positive (TP) : merupakan jumlah data *PDF Malware* yang diklasifikasikan sebagai *PDF Malware*.

False Positive (FP) : merupakan jumlah data PDF Normal yang diklasifikasikan sebagai *PDF Malware*.

True Negative (TN) : merupakan jumlah data PDF Normal yang diklasifikasikan sebagai PDF Normal.

False Negatif (FN) : merupakan jumlah data *PDF Malware* yang diklasifikasikan sebagai PDF Normal.

DAFTAR PUSTAKA

[1] I. Corona, D. Maiorca, D. Ariu, and G. Giacinto, "Lux0R," pp. 47–57, 2014, doi: 10.1145/2666652.2666657.

[2] P. Laskov and N. Šrndić, "Static detection of malicious JavaScript-bearing PDF documents," *ACM Int. Conf. Proceeding Ser.*, pp. 373–382, 2011, doi: 10.1145/2076732.2076785.

[3] M. Elingiusti, L. Aniello, L. Querzoni, and R. Baldoni, "PDF-Malware detection: A Survey and taxonomy of current techniques," *Adv. Inf. Secur.*, vol. 70, pp. 169–191, 2018, doi: 10.1007/978-3-319-73951-9_9.

[4] J. S. Cross and M. A. Munson, "Deep PDF Parsing to Extract Features for Detecting Embedded Malware," no. September, pp. 1–18, 2011, doi: 10.2172/1030303.

[5] N. K. Chauhan and K. Singh, "A review on conventional machine learning vs deep learning," *2018 Int. Conf. Comput. Power Commun. Technol. GUCON 2018*, pp. 347–352, 2019, doi: 10.1109/GUCON.2018.8675097.

[6] S. J. Khitan, A. Hadi, and J. Atoum, "PDF Forensic Analysis System using YARA," *Int. J. Comput. Sci. Netw. Secur.*, vol. 17, no. 5, pp. 77–85, 2017.

[7] B. Cuan, A. Damien, C. Delaplace, and M. Valois, "Malware detection in PDF files using machine learning," *ICETE 2018 - Proc. 15th Int. Jt. Conf. E-bus. Telecommun.*, vol. 2, pp. 412–419, 2018, doi: 10.5220/0006884704120419.

[8] N. Srndic and P. Laskov, "Detection of Malicious PDF Files Based on Hierarchical Document Structure," *Proc. 20th Annu. Netw. Distrib. Syst. Symp.*, 2013, [Online]. Available: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Detection+of+Malicious+PDF+Files+Based+on+Hierarchical+Document+Structure#0.

[9] Y. Li and B. Zhang, "Detection of SQL Injection Attacks Based on Improved TFIDF Algorithm," *J. Phys. Conf. Ser.*, vol. 1395, no. 1, 2019, doi: 10.1088/1742-6596/1395/1/012013.

[10] A. A. Awad, S. G. Sayed, and S. A. Salem, "Collaborative Framework for Early Detection of RAT-Bots Attacks," *IEEE Access*, vol. 7, pp. 71780–71790, 2019, doi: 10.1109/ACCESS.2019.2919680.

[11] A. Charim, S. Basuki, and D. R. Akbi, "Detect Malware in Portable Document Format Files (PDF) Using Support Vector Machine and Random Decision Forest," *J. Online Inform.*, vol. 3, no. 2, p. 99,

2019, doi: 10.15575/join.v3i2.196.

[12] N. Šrndić and P. Laskov, “Hidost: a static machine-learning-based detector of malicious files,” *Eurasip J. Inf. Secur.*, vol. 2016, no. 1, pp. 1–21, 2016, doi: 10.1186/s13635-016-0045-0.

[13] D. Maiorca, D. Ariu, I. Corona, and G. Giacinto, “A structural and content-based approach for a precise and robust detection of malicious PDF files,” *ICISSP 2015 - 1st Int. Conf. Inf. Syst. Secur. Privacy, Proc.*, no. April, pp. 27–36, 2015, doi: 10.5220/0005264400270036.

[14] D. Maiorca, G. Giacinto, and I. Corona, “A pattern recognition system for malicious PDF files detection,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7376 LNAI, pp. 510–524, 2012, doi: 10.1007/978-3-642-31537-4_40.

[15] M. Mursalin, Y. Zhang, Y. Chen, and N. V. Chawla, “Automated epileptic seizure detection using improved correlation-based feature selection with random forest classifier,” *Neurocomputing*, vol. 241, pp. 204–214, 2017, doi: 10.1016/j.neucom.2017.02.053.

[16] B. Sanjaa and E. Chuluun, “Malware detection using linear SVM,” *8th Int. Forum Strateg. Technol. 2013, IFOST 2013 - Proc.*, vol. 2, pp. 136–138, 2013, doi: 10.1109/IFOST.2013.6616872.

[17] C. Smutz and A. Stavrou, “Malicious PDF detection using metadata and structural features,” *ACM Int. Conf. Proceeding Ser.*, pp. 239–248, 2012, doi: 10.1145/2420950.2420987.

[18] M. Cova, C. Kruegel, and G. Vigna, “Detection and analysis of drive-by-download attacks and malicious JavaScript code,” *Proc. 19th Int. Conf. World Wide Web, WWW ’10*, pp. 281–290, 2010, doi: 10.1145/1772690.1772720.

[19] N. Fleury, T. Dubrunquez, and I. Alouani, “PDF-Malware: An Overview on Threats, Detection and Evasion Attacks,” 2021, [Online]. Available: http://arxiv.org/abs/2107.12873.

[20] H. Pareek, “Malicious Pdf Document Detection Based on Feature Extraction and Entropy,” *Int. J. Secur. Priv. Trust Manag.*, vol. 2, no. 5, pp. 31–35, 2013, doi: 10.5121/ijsptm.2013.2504.

[21] B. Kovács, F. Tinya, C. Németh, and P. Ódor, “Unfolding the effects of different forestry treatments on microclimate in oak forests: results of a 4-yr experiment,” *Ecol. Appl.*, vol. 30, no. 2, pp. 321–357, 2020, doi: 10.1002/eap.2043.

[22] H. Han, W. Y. Wang, and B. H. Mao, “Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning,” *Lect. Notes*

Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 3644 LNCS, pp. 878–887, 2005, doi: 10.1007/11538059_91.

[23] G. Hu, T. Xi, F. Mohammed, and H. Miao, "Classification of wine quality with imbalanced data," *Proc. IEEE Int. Conf. Ind. Technol.*, vol. 2016-May, pp. 1712–1717, 2016, doi: 10.1109/ICIT.2016.7475021.

[24] X. Zeng and T. R. Martinez, "Distribution-balanced stratified cross-validation for accuracy estimation," *J. Exp. Theor. Artif. Intell.*, vol. 12, no. 1, pp. 1–12, 2000, doi: 10.1080/095281300146272.

[25] N. A. Diamantidis, D. Karlis, and E. A. Giakoumakis, "Unsupervised stratification of cross-validation for accuracy estimation," *Artif. Intell.*, vol. 116, no. 1–2, pp. 1–16, 2000, doi: 10.1016/S0004-3702(99)00094-6.

[26] H. He and Y. Ma, *Imbalanced Learning: Foundations, Algorithms, and Applications*, 1st ed. Wiley-IEEE Press, 2013.

[27] Andreas C. Muller & Sarah Guido, *Introduction to Machine Learning with Python: A Guide for Data Scientists*, 1st ed. O'Reilly Media, Inc., 2016.

[28] Y. X. Chu, X. G. Liu, and C. H. Gao, "Multiscale models on time series of silicon content in blast furnace hot metal based on Hilbert-Huang transform," *Proc. 2011 Chinese Control Decis. Conf. CCDC 2011*, pp. 842–847, 2011, doi: 10.1109/CCDC.2011.5968300.

[29] S. Kilgallon, L. De La Rosa, and J. Cavazos, "Improving the effectiveness and efficiency of dynamic malware analysis with machine learning," *Proc. - 2017 Resil. Week, RWS 2017*, pp. 30–36, 2017, doi: 10.1109/RWEEK.2017.8088644.

[30] A. Ben-Hur and J. Weston, "A user's guide to support vector machines.," *Methods Mol. Biol.*, vol. 609, no. January 2010, pp. 223–239, 2010, doi: 10.1007/978-1-60327-241-4_13.