e-ISSN 2407-9103 ISSN 1412-2782
Vol 23. No. 1 Januari 2024

Jurnal
POLI-TEKNOLOGI

Designing A Pdf Malware Detection System Using Machine Learning

Salman Abdul Jabbaar Wiharja[a*], Deden Pradeka, Wirmanto Suteddy
Teknik Komputer, Universitas Pendidikan Indonesia, Bandung, Indonesia
*email of coresponding author: salmanfc207@upi.edu

Article History
Submitted: 11/01/2024
Revised : 03/02/2024
Accepted : 03/02/2024
Published: 22/02/2024

Keywords:
Machine Learning, PDF, Malware, Security, Computer, Random Forest

Abstract

This research proposes an approach to build malicious PDF detection system using random forest algorithm, focusing the Evasive-PDFMal2022 dataset which is updated and extended with the addition of new datasets. This dataset includes malicious PDF files from CVE and Exploit-DB, non-malicious PDF files, as well as files from private collections and Technically-oriented PDF Collection. Features were extracted using the PDFID tool, resulting in 29 structural features that formed the basis for the Random Forest classification algorithm. Experiments showed that the model trained with the new dataset provided accuracy equivalent to the Evasive-PDFMal2022 model, at 98%, albeit with a small decrease in recall for the benign class. In addition, this research involved the creation of a website for metadata extraction and malicious PDF detection. Recognition goes to the dataset contributors, tool developers, and dataset providers from NIST and Exploit-DB. Overall, this research successfully increased the representation and diversity of the dataset, provided good model training results, improved detection from 3 malicious PDF variants to 13 variants, and created a practical tool for malicious PDF extraction and detection. Nonetheless, further development may be required to improve detection performance in more complex scenarios.

Introduction

The Portable Document Format (PDF) has gained immense popularity owing to its versatile integration of diverse content, including graphics, videos, images, and various data types [1]. In recent years, the Portable Document Format (PDF) has established itself as the preferred standard for document exchange and dissemination. It is widely acclaimed for its adaptability, customizable features, and effortless portability across various platforms. [2]. Nevertheless, the pervasive use of PDFs has attracted the attention of cyber attackers aiming to exploit vulnerabilities and manipulate file features, thereby evading established security measures [3].

A comprehensive analysis of global telemetry data, gathered from Palo Alto Networks Next-Generation Firewall (NGFW), Advanced URL Filtering, Cortex Data Lake, and Advanced Wildfire spanning from 2019 to 2022, reveals a significant surge in vulnerability exploitation. In 2019, 45,492 instances were recorded, escalating to 128,063 in 2020, 147,342 in 2021, and a staggering 228,345 in 2022 [4] Notably, PDF emerges as the most favored file type employed as a malicious email attachment, constituting 66% of such attachments. Given the prevalence of PDF usage in business environments, victims may be less vigilant when encountering expected file types, as opposed to the wariness elicited by unexpected file formats like EXE. Compounded by a lack of awareness regarding the malicious potential of PDFs and their ability to elude modern antivirus detection, PDFs have become a favored vector for cyber threats [5].

Within this landscape, malicious documents stand out as a prominent method employed by attackers to propagate malware. Malicious code finds its way into PDFs through various means, including encryption flows, executable files (exe), JavaScript, system commands, and hidden objects [6]. The intricate nature of the PDF file format, accommodating diverse content such as scripts, forms, and multimedia elements, poses a considerable challenge in detecting malicious content. This challenge is exacerbated when attackers employ evasive techniques to conceal their malicious payloads [2]. Consequently, PDFs have become an effective tool for launching social engineering attacks, presenting a heightened risk for the proliferation of malware [5].

Theory

A. Structure of PDF

PDF (Portable Document Format) has transformed into a versatile solution designed for seamless sharing of diverse content, including text, rich media, and images. Its capability extends without being confined to specific hardware or software platforms. Introduced by Adobe in 1993, PDF has undergone significant evolution and has universally been embraced as a standard for document sharing, achieving standardization as an open standard by ISO [2].

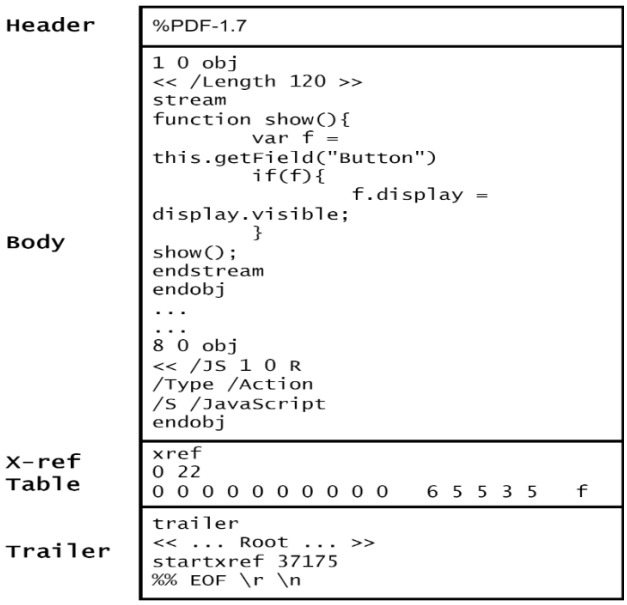

Figure 1. Structure of Portable Document File

The fundamental structure of a PDF document [7], [8], [9], [10] as illustrated in Figure 1, comprises four primary components:

1. **Header**
   Information regarding the standard PDF version's format is provided in the Header section; the standard PDF format is recorded in ISO and needs to match the format. For instance, "%PDF-1.1." is used to indicate the header of a PDF version 1.0 document, and "%PDF-1.1." is used to indicate a PDF version 1.7 document. The header must appear in the first 1024 bytes of the file in order for the Acrobat Reader to function.
2. **Body**
   The body encompasses user-visible content, defining file operations through objects. It incorporates embedded data like text, images, and script codes, presented as objects. Operations like data decompression or decryption, if necessary, are detailed within these objects and are typically executed during file rendering.
3. **Cross-reference (X-ref)**
   Outlines the offset of each object to be rendered by the reader application within the PDF file. Adhering to established PDF specifications, the x-ref table's offsets enable random access to any object in the file, facilitating incremental modifications to the document. As updates occur, additional x-ref tables and trailers are appended to the end of the document.
4. **Trailer**
   This dedicated segment resides at the conclusion of the file. It furnishes details regarding the object identified by the /Root tag, which serves as the document viewer's initial object. The trailer encapsulates the file's last line, denoted by '%%EOF'.

In other words, each indirect object that the x-ref table refers starts to be parsed at the trailer object when the file is shown in a PDF reader. Simultaneously, the reader decompresses the data gradually, rendering all elements—text, images, and other components of PDF files—gradually. The way the structure is organized highlights how the PDF file functions as an object graph, providing guidance to the PDF reader and controlling how the user is presented with the contents of the file [9].

B. Malware in PDF

The inherent versatility of the Portable Document Format (PDF), designed to seamlessly incorporate diverse content, not only facilitates legitimate information sharing but also creates opportunities for attackers to exploit potential vulnerabilities. Malicious programs adeptly conceal their presence, employing a variety of sophisticated methods, including histogram and pixel pattern-based techniques, as detailed in [11]. Subsequent research by [12] delves into the utilization of steganography to discreetly embed malicious code within images, employing diverse means. Furthermore, [13] underscores the effectiveness of cryptography, utilizing substitution and permutation, as a means to conceal and safeguard malicious data within images.

In the realm of PDF files, the exploitation of vulnerabilities extends beyond the utilization of JavaScript code, encoded streams, and embedded objects such as executable files, ShockWave Flash (SWF), and image files. Specific PDF tags pose inherent risks and have the potential to cause harm, as noted in [14] These tags include:

- /JS and /JavaScript : JavaScript scripts in these tags can be employed to open a backdoor.
- /AA and /OpenAction : Automated actions within the PDF can be initiated.
- /GoTo : Tags facilitating movement to a specific page within or outside the PDF document.
- /Launch : Can open/launch a document or run a program.
- /URI : Enables access to a URL.
- /SubmitForm and /GoTo : Facilitate data submission to the specified URL in the PDF document.
- /RichMedia : A tag for embedding Flash in PDF documents.
- /ObjStm : A tag capable of concealing the Object Stream.

Mendemonstrasikan perilaku berbahaya yang melampaui batasan tag PDF, contoh seperti eksploitasi kerentanan korupsi heap Reader BMP/RLE (CVE-2013-2729) dan eksekusi kode biner melalui visualisasi file PDF (CVE-2010-1240) menunjukkan kerentanan keamanan yang berimplikasi luas. Analysts commonly employ keyword-based analysis to discern potential malware, searching for terms such as URI, /RichMedia, /JavaScript, /OpenAction, and /GoTo. Absence of these keywords may lead to the classification of the file as benign [5]. The tool PDFiD [3], [15], [16], [17] is employed for textual analysis of all dictionaries within a PDF file, including those not explicitly provided by reader software.

C. Machine Learning Classification

Machine Learning (ML) is a broad category of algorithms with applications ranging from sentiment analysis of YouTube comments to handwriting recognition [18] using methods like Support Vector Machines (SVM) and Random Forest [19]. Machine learning (ML) is a key player in the field of classifying harmful PDFs. It plays a major role in identifying and detecting dangerous elements present in PDF files.

Previous studies, such as the one by [7] explore the field of group learning techniques. Effective data classification methods are investigated, including Random Forest, Random Subspace, AdaBoost, Stacking, and Random Committee. As an example, [20] uses Convolutional Neural Networks (CNN) to classify harmful PDFs, and [19], uses the stacking learning technique to increase classification accuracy.

Random Forest is a standout algorithm among the ensemble learning techniques. It combines predictions from many decision models and leverages the power of multiple decision trees cooperating [21], [22]. Random Forest's dependability is applicable in a variety of settings, such as the Internet of Things (IoT) [22]. Previous research [23], [24], [25], [26], [27], has shown that the Random Forest algorithm is reliable when it comes to classification; nevertheless, this study specifically highlights the Random Forest algorithm's use in the context of harmful PDF classification.

D. Machine Learning Website

The development of a website for PDF malware classification, incorporating machine learning, demands a nuanced and multidisciplinary approach. This intricate endeavor necessitates proficiency in web programming, cryptography, and the implementation of machine learning algorithms. As emphasized in [28], the initial step involves gaining expertise in fundamental web programming languages such as HTML, CSS, and JavaScript to construct a responsive user interface. Additionally, the incorporation of cryptography within websites becomes crucial to safeguard sensitive data, including the machine learning models utilized in the classification process. A comprehensive understanding of the application of cryptography in website development is elucidated in the references cited [28].

The paper [29] offers insightful information on applying machine learning in classification settings within the field of machine learning. The concepts discussed in this article can be easily modified to categorize PDF viruses. As a result, careful feature selection, data processing, and rigorous model testing are necessary for the integration of machine learning models.

Moreover, the article [30] contributes a pragmatic outlook on leveraging the Python programming language for website design analysis. The adoption of Python proves instrumental in facilitating the implementation of machine learning models and their seamless integration into the website infrastructure. Therefore, choosing the Python programming language for developing a website dedicated to PDF malware classification not only provides flexibility but also enhances overall efficiency. This strategic choice aligns with the intricacies of implementing machine learning within the web framework, ensuring a harmonious integration that streamlines the classification process and enhances the user experience.

Method

This section delineates the methodology employed by the author to construct an effective system for detecting malicious PDFs using machine learning. The foundation of this approach relies on an enriched dataset comprising 20 features and meticulously labeled data, totaling 12,600 rows, which forms the core of the training set. The next phase involves applying the Random Forest classification algorithm to train the Evasive-PDFMal2022 and NewDatasets. In this process, the selected features are fed into the algorithm, enabling it to discern distinctive traits between benign and malicious PDF files. As a result, the system evolves to proficiently predict and classify unlabeled PDF files, accurately determining their safety status. The comprehensive schematic of this method is visually represented in Figure 2.

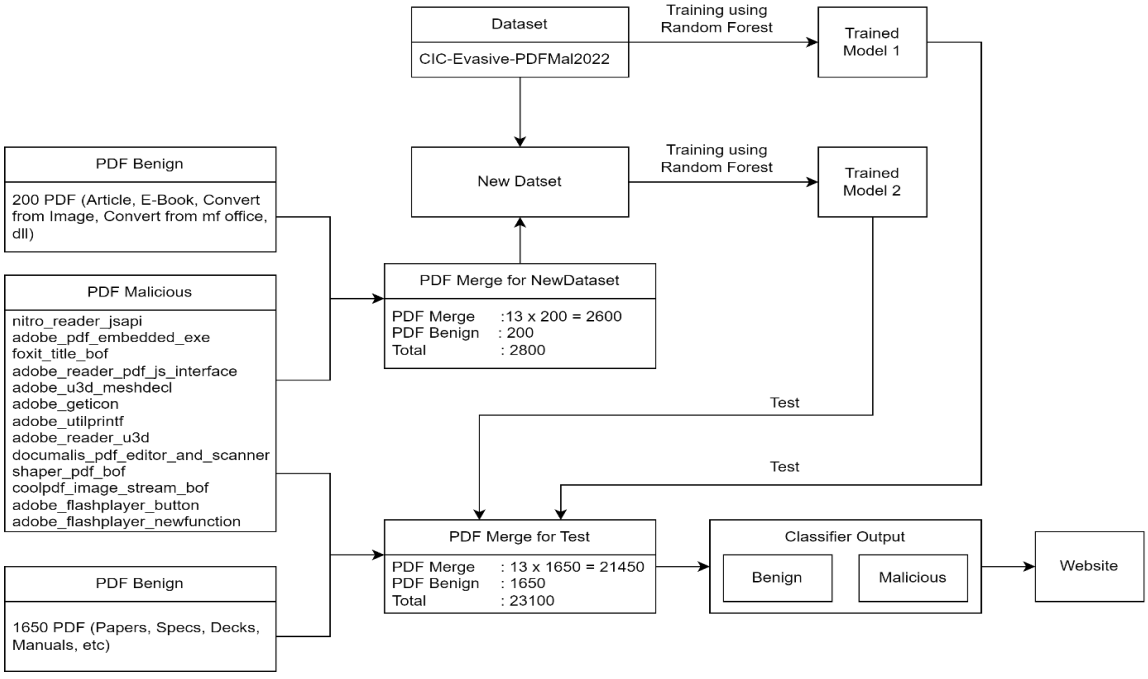

Figure 2. Illustrates the proposed enhanced dataset-based approach.

The development of the PDF classification system follows a systematic sequence of steps, elaborated upon in the following subsections:

A. Dataset

Evasive-PDFMal2022: The foundation of the PDF classification system lies in the Evasive-PDFMal2022 dataset, introduced by [17]. A significant improvement over the Contagio PDF dataset, Evasive-PDFMal2022 addresses identified weaknesses, including a high prevalence of duplicate samples (up to 44% of the total dataset) and insufficient sample diversity within each class. Comprising 10,025 samples categorized into 4,468 benign and 5,557 malicious files, this enhanced dataset aims to offer a more realistic representation of PDF distribution. By eliminating duplicates and ensuring diverse samples within each class, these enhancements bolster the dataset's validity and representativeness, vital for robust research in malicious PDF detection.

B. PDF Collection

In addition to Evasive-PDFMal2022, two supplementary dataset collections contribute to the research: Private PDF Collection and Technically-oriented PDF Collection.

1. **Private PDF Collection**
   Comprising 200 original PDF files spanning various content types such as articles, e-books, and converted PDFs, this collection undergoes data extraction. Merging the results with Evasive-PDFMal2022 yields a richer, more diverse dataset.
2. **Technically-oriented PDF Collection**
   Curated by tpn [31] in their GitHub repository, this collection includes 1,650 PDF files encompassing technical documents like papers, specifications, presentations, and manuals. Digunakan sebagai dataset pengujian, evaluasi ini menilai kapasitas model untuk menggeneralisasi seluruh dokumen teknis. Penggabungan Evasive-PDFMal2022 dengan Koleksi PDF Pribadi memperkaya keterwakilan dataset, dan penyertaan Koleksi PDF yang berorientasi teknis memberikan tolok ukur yang berharga untuk mengukur kinerja model.

C. Feature Extraction

The proposed detection system relies on 21 structural features, extracted using the latest variant of PDFMalyzer, an open-source software available at [32]. Derived from PDFID [15] and PyMuPDF [33] , PDFMalyzer extracts structural features crucial for distinguishing non-malicious PDF files. Table I presents a comprehensive overview of the initial feature set, encompassing 21 distinctive structural features that play a crucial role in the classification process.

TABLE I. OVERVIEW OF THE INITIAL FEATURE SET COMPRISING 21 FEATURES

Feature Name	Description of Feature
Obj	opening objects tags found
Endobj	object closers
Stream	stream openers
Endstream	stream closers
Xref	X-ref (Cross ref) tables in PDF
Trailer	trailers in PDF
Startxref	start X-refs indicators in PDF
/page	pages in PDF
/encrypt	Indicates wheather the document has a Password

/Objstm	Objstm (object streams) in PDF
/JS	JS in PDF
/JavaScript	javascript in PDF
/AA	Automatic actions in a single event in PDF
/OpenAction	Number of automatic actions when a PDF is opened
/Acroform	Number of acrobat forms
/JBIG2Decode	Whether the document is compressed with JBIG2
/RichMedia	Number of embedded media in PDF
/launch	Number of actions in PDF
/EmbeddedFile	Number of Embedded keywords found in PDF
/XFA	Number of XML keywords in PDF
/Colors	Number of colors present in PDF

This initial set of structural features is comprehensive, encompassing distinctive characteristics uncommon in ordinary PDF files. Their presence or absence serves as a crucial indicator for determining the malicious or benign nature of a PDF file. The meticulous extraction and analysis of these structural features lay the foundation for a robust detection system capable of discerning between benign and malicious PDF file.

D. Creating a New Dataset

The Evasive-PDFMal2022 dataset, which was carefully examined by [17], as well as an expansion of the Contagio dataset, demonstrated certain shortcomings in terms of successfully countering the different kinds of PDF malware listed in the Exploit-db and Common Vulnerabilities and Exposures (CVE) of the National Vulnerability Database (NIST). In order to fully address the range of PDF-based risks in the cybersecurity space, the authors of this research started the process of developing a new dataset.

1. Dataset Methodology
    a. **Extraction of Metadata from Benign PDFs**
    The process began with the extraction of metadata from an initial set of 200 original PDFs carefully selected to represent diverse content types, ensuring a robust foundation for the dataset.
    b. **Integration of Malicious PDFs**
    Subsequently, 13 malicious PDFs, sourced from the Common Vulnerabilities and Exposures (CVE) and Exploit-db repositories, were strategically integrated. Each malicious PDF represented a unique exploit, covering distinct vulnerabilities and exploitation techniques.
2. Merging Process

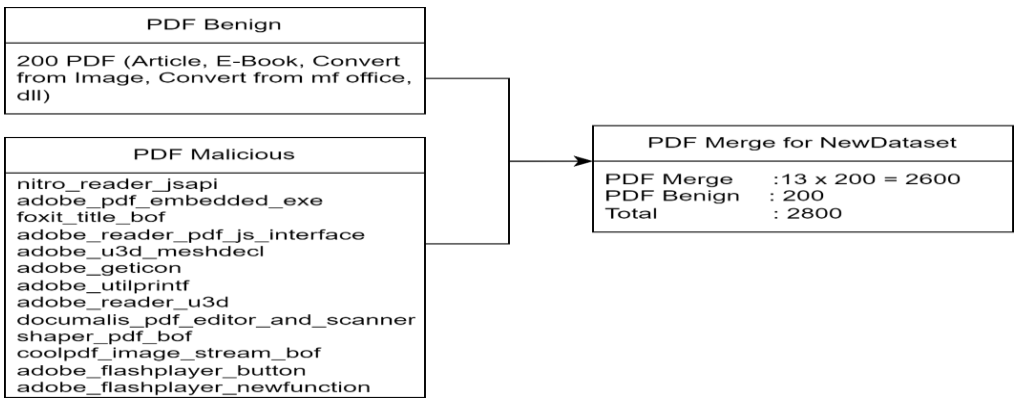

Figure 3. Process of Merging Malicious Pdfs With Non-Malicious Pdfs

Our methodology aligns with the insights from [34], emphasizing the profound impact of training data volume on model accuracy. In response, the author intentionally duplicates each type of malicious PDF to augment the training data, aiming to enhance test results. Figure 3 provides a detailed, step-by-step illustration of the merging process, depicting how each type of malicious PDF seamlessly integrates with a benign original PDF. This visual representation emphasizes our meticulous consideration for dataset size, showcasing our commitment to a comprehensive and representative training set. Aligned with contemporary research practices, this approach underscores our dedication to optimizing the model's performance through purposeful dataset expansion.

3. Selected Malicious PDFs

The malicious PDFs selected for integration include:

a. Nitro Reader JS API (CVE-2017-7442)
This exploit targets a vulnerability in the JavaScript API of Nitro and Nitro Pro PDF Reader version 11, allowing unauthorized writing of arbitrary files and local file execution, posing a severe security risk.
b. Adobe PDF Embedded EXE (CVE-2010-1240)
Involves inserting a Metasploit payload—intended for possible social engineering attacks—into an already-existing PDF file. The resultant PDF can be circulated deceitfully and used as a weapon.
c. Foxit Title BOF (EDB-ID-16621)
This vulnerability is exploited when a malicious PDF file with an unusually long string in the Title field is opened in Foxit PDF Reader (pre-version 4.2.0.0928). Records pertaining to structured exception handling are overwritten as a result of the overflow. Most notably, JavaScript is not needed for this exploitation.
d. Adobe Reader PDF JS Interface (CVE-2014-0514)
This module is designed to target Adobe Reader versions lower than 11.2, offering an insecure native interface to untrusted JavaScript embedded within PDFs. The module incorporates the android/webview_addjavascriptinterface browser vulnerability into a PDF, with the objective of gaining command shell access.
e. Adobe U3D Meshdecl (CVE-2009-3953)
Utilizes Acrobat and Adobe Reader versions less than 7.1.4, 8.2, and 9.3 in order to exploit an array overflow. An attacker can create a PDF with damaged U3D data and leverage the array overflow vulnerability to execute any code.
f. Adobe Geticon (CVE-2009-0927)
uses Adobe Acrobat and Reader versions less than 7.1.1, 8.1.3, and 9.1 to exploit a buffer overflow. The attack involves creating a specifically created PDF with a broken Collab.getIcon() call, which allows arbitrary code to be executed.
g. Adobe Utilprintf (CVE-2008-2992)
takes advantage of a buffer overrun in Acrobat Professional and Adobe Reader versions older than 8.1.3. An attacker can create a PDF with a broken util.printf() element and use that PDF to run arbitrary code. This is the methodology used for the hack.
h. Adobe Reader U3D (CVE-2011-2462)
Uses an uninitialized memory vulnerability to target Adobe Reader. Creates a PDF document that contains carefully designed U3D data, which allows for arbitrary code execution. JavaScript's heap spray technique guarantees control over memory used by invalid pointer problems.
i. Documalis PDF Editor and Scanner (CVE-2020-7374)
The software's failure to validate the contents of JPEG images stored within PDF files leads to a stack-based buffer overflow. A successful exploitation gives the attacker remote code execution when the software is run.
j. Shaper PDF BOF (EDB-ID-37760)
Security flaws in PDF Shaper are visible when processing PDF files, particularly when converting PDFs to images. Specially created PDF files can be exploited. The module has been successfully tested on Windows XP, 7, 8, and 10.
k. Coolpdf Image Stream BOF (CVE-2012-4914)
This vulnerability manifests as a susceptibility that, upon viewing a malicious PDF file containing a custom image stream, can be exploited to trigger a stack buffer overflow in Cool

PDF Reader (version < 3.0.2.256). Successful validation of this exploit has been conducted on Cool PDF 3.0.2.256, passing testing on Windows XP (SP-3) and Windows 7 (SP-1).

l. Adobe Flashplayer Button (CVE-2010-3654)
focuses on a flaw in the way Adobe Flash Player versions 9.x.x and 10.0.x handle specific SWF movies. Applications that embed the Flash player, such as Adobe Reader and Acrobat, are likewise susceptible. enables the embedding of carefully designed Flash movies into PDF documents, resulting in arbitrary code execution. uses a DEP bypass technique akin to that of the adobe_libtiff module to enable controlled memory execution via the AcroJS stack spray.

m. Adobe Flashplayer Newfunction (CVE-2010-1297)
Uses specifically designed Flash movies included in PDF documents to take advantage of a flaw in the way Adobe Flash Player versions 9.x.x and 10.0.x handle some SWF movies. This allows for arbitrary code execution. For controlled memory execution, utilize the AcroJS stack spray.

4. Resulting Dataset
After merging and metadata extraction, a total of 2,800 PDFs were generated, comprising 2,600 malicious and 200 benign samples. PDFID [17] was then employed for additional metadata extraction, and the resulting information was merged with the Evasive-PDFMal2022 dataset, resulting in the creation of a comprehensive dataset referred to as NewDataset. The cumulative dataset utilized in this study encompasses a total of 12,825 sample PDF files, comprising 4,668 benign and 8,157 malicious instances.

E. Dataset Training

Moving to the training phase, the author provides a concise overview of the dataset training process within the proposed system. The Random Forest algorithm was chosen for classification, utilizing the basis of Random Tree classification [7]. The training process encompasses two main stages: using the Evasive-PDFMal2022 dataset and incorporating the NewDataset proposed by the author. The latter entails an addition of 2800 rows of data, as detailed in Figure 3. Further discussion on the training results will be explored in the Results and Discussion section.

F. Website Creation

Post the dataset training and evaluation stages yielding optimal accuracy, the author proceeds with the creation of a website. This website is envisioned to facilitate PDF metadata extraction and malicious PDF detection, serving as a valuable tool for users in identifying and managing PDF files with potential security risks. The website creation marks a crucial step in extending the utility of the developed system, enhancing user support for cybersecurity efforts.The ensuing sections will delve into a comprehensive analysis of the dataset training results and an in-depth discussion of the created website within the Results and Discussion segment.

Results and Discussion

A. New Dataset Creation

The critical role of advanced and diverse datasets in fortifying machine learning models has been consistently emphasized in prior research [35]. The intrinsic link between dataset quality and the effectiveness of detection outcomes is highlighted by [36]. In this segment, a careful examination is conducted on the experimental outcomes, scrutinizing the impact of creating novel datasets on the Random Forest classifier's performance. The emphasis is specifically directed towards addressing the inherent constraints of the Evasive-PDFMal2022 dataset.

Various concealment methods, including steganography techniques, can be employed to hide information in PDF files, as elucidated in previous research [37]. In this study, the authors utilized tools such as pdftk and mergepdf to amalgamate PDFs, thereby creating a NewDataset through the fusion of malicious PDFs with non-malicious counterparts, as depicted in Figure 4.

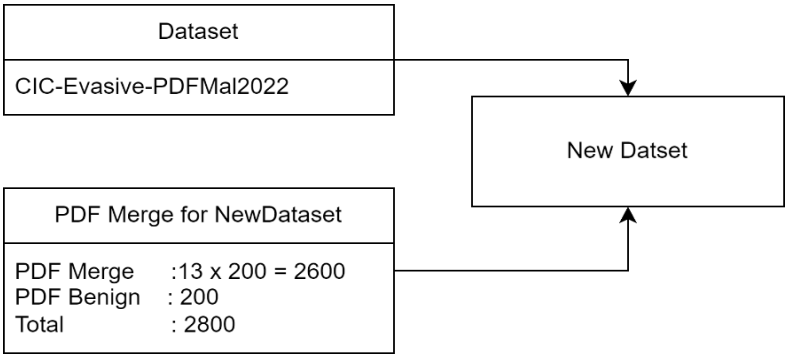

Figure 4. Dataset Merging Process To Generate Newdataset

The merging process involves the consolidation of 13 malicious PDF files sourced from the CVE and Exploits-db datasets with 200 Private PDF Collection files. The outcome yielded 2600 PDFs labeled as "Malicious" and 200 PDFs labeled as "Benign." Metadata extraction, facilitated by the PDFID tool and enriched with Python scripts, produced CSV-formatted output. Subsequently, class labels based on characteristics were incorporated, and the dataset was seamlessly integrated with Evasive-PDFMal2022.

This strategy aims to bolster the model's dependability while maintaining classification accuracy when integrated with established datasets. The subsequent section delves into a comprehensive analysis of how the introduction of the NewDataset influences the overall performance of the Random Forest classifier.

B. Comparing Training Results

The original Evasive-PDFMal2022 and the enhanced NewDataset were two different datasets used in a crucial experiment to train the Random Forest classifier model. The NewDataset is an amalgam of metadata taken from the combination of malicious and benign PDFs, along with Evasive-PDFMal2022. This analysis's main goal is to evaluate in-depth how the addition of the new dataset affects the model's overall performance.

1. Hyperparameter Configuration

Before delving into the training results, it is crucial to outline the hyperparameter configuration used in fine-tuning the Random Forest classifier. The choice of hyperparameters significantly influences the model's behavior and its ability to generalize to unseen data. In this experiment, careful consideration was given to defining a set of hyperparameters that strikes a balance between model complexity and robustness for the task of detecting malicious PDF files. The hyperparameter values used during Random Forest classifier training can be seen in table II below:

TABLE II. RANDOM FOREST CLASSIFICATION RESULTS WITH EVASIVE-PDFMAL2022 DATASET

Hyper-parameter	Value
n_estimators	100
min_samples_leaf	1
min_samples_split	2
bootstrap	True
max_depth	None

2. Training Results with Evasive-PDFMal2022 dataset

In Table III, the cross-validation results of the Random Forest classifier with the Evasive-PDFMal2022 dataset are showcased. Despite maintaining a high accuracy of 98%, a marginal decrease in benign recall indicates potential unreliability in detecting benign files.

TABLE III. RANDOM FOREST CLASSIFICATION RESULTS WITH EVASIVE-PDFMAL2022 DATASET

	Precision	**Recall**	**F1-Score**
Benign	98 %	98 %	98 %
Malicious	99 %	99 %	99 %
Accuracy			98 %

In Figure 5, the outcomes of the confusion matrix illustrate the training results using the Random Forest algorithm with hyperparameters, as previously described. The model achieved a notable True Positive (TP) value of 3859 instances, signifying the accurate identification of malicious PDFs. Additionally, a True Negative (TN) count of 1588 instances indicates the correct recognition of benign files. However, the model encountered challenges, as reflected in the relatively high number of False Positive (FP) instances, totaling 17,604. These instances represent benign files incorrectly classified as malicious. Furthermore, the model encountered a minimal number of False Negative (FN) instances, totaling 62, depicting malicious files that were misclassified as benign. This detailed analysis offers a thorough comprehension of the model's strengths and identifies areas for improvement. These insights are vital for refining the algorithm and bolstering its performance in real-world scenarios.

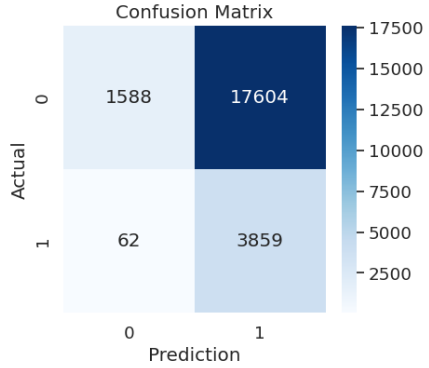

Figure 5. Confusion matrix of Training Results with Evasive-PDFMal2022 dataset

3. Training Results with NewDataset

Table IV displays the cross-validation results of the Random Forest classifier with NewDataset, which combines Evasive-PDFMal2022 with the new dataset. Despite maintaining high accuracy (98%), a slight decrease in benign recall suggests potential unreliability in detecting benign files.

TABLE IV. RANDOM FOREST CLASSIFICATION RESULTS WITH NEW DATASET

	Precision	**Recall**	**F1-Score**
Benign	98 %	97 %	98 %
Malicious	99 %	99 %	99 %
Accuracy			98 %

Figure 6 encapsulates the training outcomes using the Random Forest algorithm with the hyperparameters previously outlined, applied to the NewDataset. The results showcase a robust performance, highlighted by a substantial True Positive (TP) count of 21,308 data points, indicative of successful identification of malicious PDFs. The True Negative (TN) value of 1105 instances underscores the model's proficiency in correctly recognizing benign files. However, there are nuanced challenges apparent in the 155 instances of False Positive (FP), where benign files were misclassified as malicious, and the 545 instances of False Negative (FN), representing malicious files mistakenly identified as benign. This detailed examination provides valuable insights into the model's efficacy, guiding further refinement for heightened accuracy and reliability in practical applications.

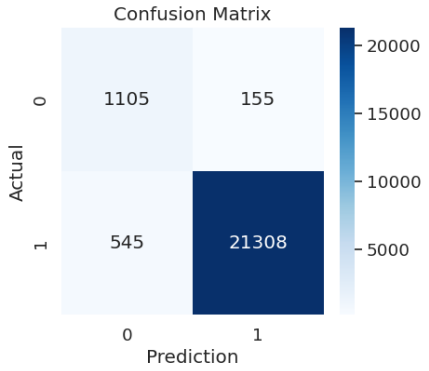

Figure 6. Confusion matrix of Training Results with NewDataset

4. Effect of New Dataset on Malicious PDF Variants

Figure 7 presents a comparative analysis of the Random Forest model's training accuracy in identifying diverse malicious PDF variants. The model trained with the NewDataset exhibits superior performance, achieving heightened accuracy in recognizing a spectrum of 13 malicious PDF types. This stands in stark contrast to the Evasive-PDFMal2022-trained model, which exhibited proficiency in identifying only 3 types. Despite a marginal decrease in benign recall, this compromise is deemed acceptable in light of the substantial advancements made in the detection of diverse malicious PDF variants.

The observed decrease in benign file detection accuracy can be attributed to the significant skew in the training data, which is heavily dominated by malicious files. As a result, the model excels in reliably identifying malicious files but may exhibit reduced accuracy in detecting benign ones. This trade-off is a strategic decision made to enhance the model's effectiveness in the primary objective of identifying and mitigating malicious PDFs, acknowledging the inherent imbalance in the dataset composition.

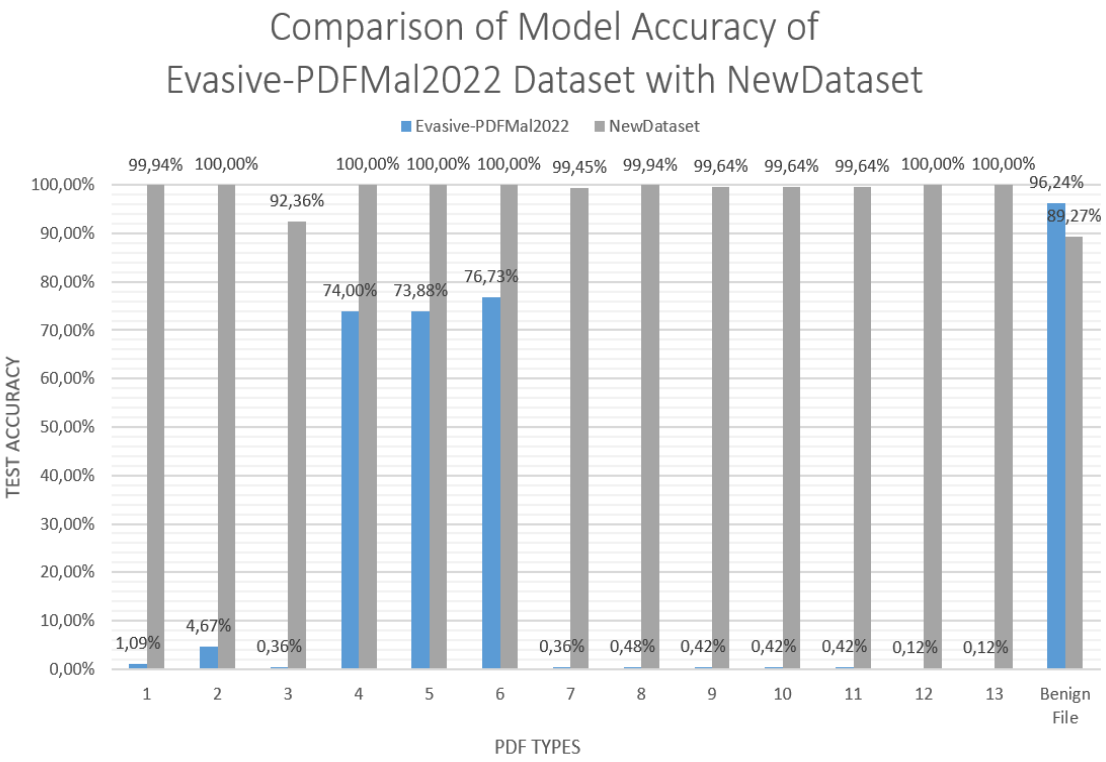

Figure 7. Accuracy Comparison of Random Forest Model Training Results.

In Upon close scrutiny of Figure 7, where the model's performance is tested on the merged PDF results, nuanced insights come to light. The model exhibits a relatively modest accuracy, particularly in recognizing specific malicious PDF variants (4, 5, and 6), namely adobe_reader_pdf_js_interface, adobe_u3d_meshdecl, and adobe_geticon, with an accuracy exceeding 73%. Interestingly, the model excels in accurately identifying benign files, boasting an accuracy surpassing 96%.

This contrast is stark when compared to the model trained with the new dataset, showcasing an exemplary accuracy of 99% in detecting malicious PDF files, highlighting a remarkable enhancement. Despite a slight dip in accurately identifying non-malicious files to approximately 89.27%, this remains within an acceptable margin. The noticeable progress in detecting various malicious PDF variants stands as a testament to the efficacy of the new dataset.

It's essential to note that the elucidation of PDF Types is provided in the Method section, particularly in the segment covering Selected Malicious PDFs. This improvement is particularly noteworthy when juxtaposed against the limitations of the previous model, which could only recognize three types of malicious PDFs. With the integration of the new dataset, the model now adeptly identifies 13 types of malicious PDFs, showcasing its heightened sensitivity to diverse variants. While there's a marginal dip in non-malicious file detection, this concession is inconsequential, considering the primary focus on identifying malicious PDFs.

The substantial leap in accuracy underscores the significance of crafting sophisticated datasets, reinforcing the notion that the quality of training data profoundly impacts the robustness and versatility of machine learning models. This enhancement bodes well for the overarching goal of effective and nuanced malicious PDF detection

C. Creating a Website

In the final phase of the research, the author created a website with the intention of delivering future benefits. The interface of the malware detection web is illustrated in Figure 8.

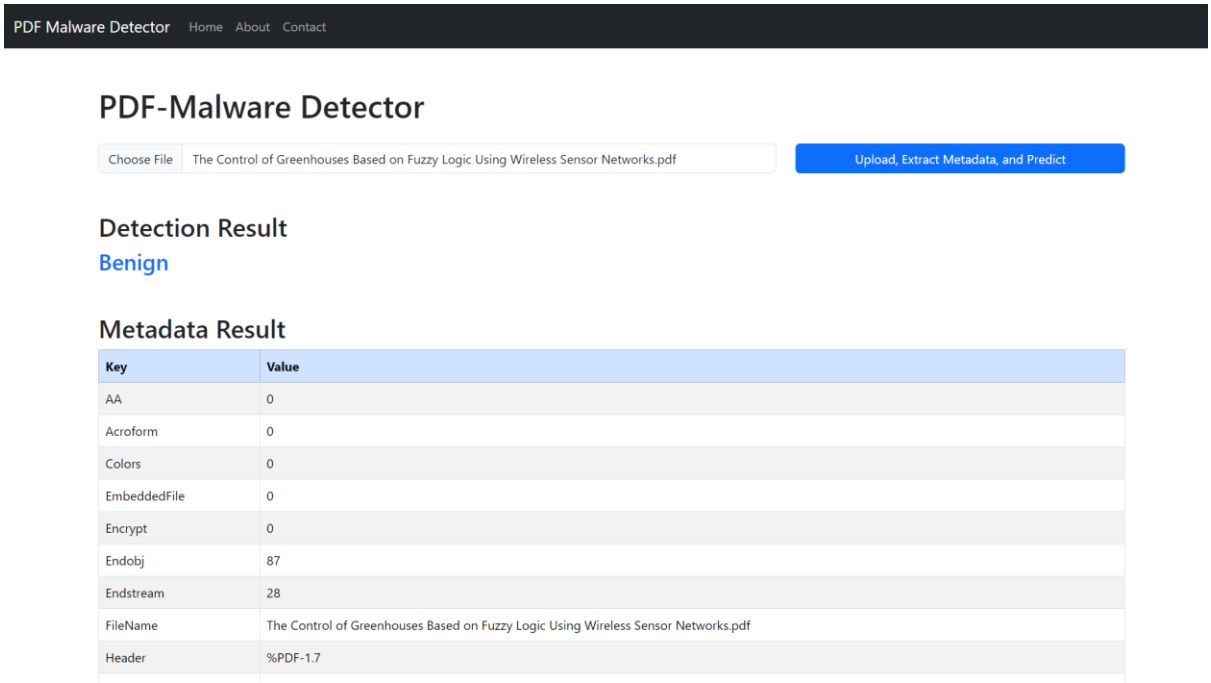

Figure 8. PDF Malware Detection and Data Extraction Website

As shown in Figure 8, the website facilitates metadata extraction from uploaded PDFs and provides classification results on the web interface. Users receive information on whether the file falls into the "Malicious" or "Benign" category.

This website represents a crucial tool in supporting users in identifying and managing PDF files with potential security risks. The subsequent sections will delve into a comprehensive analysis of the dataset training results and an in-depth discussion of the created website in the Results and Discussion segment.

Conclusion

This research endeavors to fortify the landscape of malicious PDF detection through a meticulously crafted approach rooted in machine learning. Leveraging the Evasive-PDFMal2022 dataset as a foundation, the authors extend its capabilities by introducing a novel dataset. This amalgamation encompasses a diverse array of PDFs, including those sourced from CVE and Exploit-db, private collections, and the Technically-oriented PDF Collection. Employing PDFID, a potent open-source tool, the research extracts 29 structural features crucial for robust classification.

Algoritme klasifikasi yang dipilih, Random Forest, menjalani pelatihan yang ketat dengan menggunakan dataset Evasive-PDFMal2022 yang sudah ada dan dataset NewDataset yang baru saja dikurasi. Temuan eksperimental mengungkapkan kesetaraan yang luar biasa dalam akurasi antara kedua model, dengan tingkat akurasi yang mengesankan, yaitu 98%. Meskipun ada sedikit penurunan dalam penarikan kelas jinak yang diamati, yang disebabkan oleh ketidakseimbangan yang melekat dalam volume data pelatihan kelas jinak dan berbahaya, jalan untuk eksplorasi di masa depan disarankan - mengupayakan distribusi yang lebih seimbang dalam kumpulan data pelatihan. Despite this nuanced challenge, the overall prowess of the model in proficiently classifying PDF files remains commendable. Notably, when subjected to the amalgamated data from CVE, Exploit-db, and Technically-oriented PDF Collection, the model trained with the authors' innovative dataset markedly amplifies its detection capabilities—from discerning three malicious PDF variants to a formidable 13 variants.

In tandem with model refinement, the research culminates in the creation of a purposeful website. This online tool, adept at PDF metadata extraction and malicious categorization, emerges as a practical asset for users keen on assessing the security integrity of uploaded PDF files.

In summation, this research triumphs in augmenting dataset representation and diversity for heightened malicious PDF detection efficacy. The adept model training results and the practical utility of the website underscore the tangible impact of this endeavor. However, it's imperative to acknowledge the inherent imperfections of any system, emphasizing the ongoing need for nuanced development to enhance detection performance in intricate scenarios.

Acknowledgment

We extend our sincere gratitude to the researchers behind the Evasive-PDFMal2022 dataset, especially Issakhani et al. Their generosity in providing this foundational dataset has been instrumental in advancing our research and enhancing methods for detecting malicious PDFs.

Our appreciation also goes to the open-source community, particularly the developers of PDFMalyzer and PDFID. These invaluable tools have played a significant role in extracting features from PDF files, forming the backbone of our research methodology.

We acknowledge the National Vulnerability Database (NIST) and Exploit-db for their pivotal role in providing the dataset of malicious PDFs that fueled the development of the NewDataset. Their commitment to offering comprehensive information on vulnerabilities and exploits has significantly enriched the representativeness of our dataset.

We especially thank the developers of several open-source programs that were essential to our research as well as the teamwork that went into creating the Random Forest algorithm. We would like to express our gratitude to everyone who so kindly shared their knowledge, given resolute support, and supplied technical help during the challenging stages of this study.

This collective collaboration underscores the essence of community-driven research, and we express our deepest thanks to all those who have been part of this journey.

References

[1] H. Bae, Y. Lee, Y. Kim, U. Hwang, S. Yoon, and Y. Paek, "Learn2Evade: Learning-Based Generative Model for Evading PDF Malware Classifiers," *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 4, pp. 299–313, Aug. 2021, doi: 10.1109/tai.2021.3103139.

[2] International Organization for Standardization, *ISO 32000-2:2020 (PDF 2.0)*, 2nd ed. Switzerland: PDF Association, Inc., 2020.

[3] N. Nissim, A. Cohen, C. Glezer, and Y. Elovici, "Detection of malicious PDF files and directions for enhancements: A state-of-the art survey," *Computers and Security*, vol. 48. Elsevier Ltd, pp. 246–266, Feb. 03, 2015. doi: 10.1016/j.cose.2014.10.014.

[4] Paloalto Networks, "Network Threat Trends Research Report," 2023.

[5] P. Singh, S. Tapaswi, and S. Gupta, "Malware Detection in PDF and Office Documents: A survey," *Information Security Journal*, vol. 29, no. 3. Taylor and Francis Inc., pp. 134–153, May 03, 2020. doi: 10.1080/19393555.2020.1723747.

[6] ilovepdf.com, "Top tips for protecting your PDFs," iLovePDF - Online tools for PDF.

[7] S. Y. Yerima and A. Bashar, "Explainable Ensemble Learning Based Detection of Evasive Malicious PDF Documents," *Electronics (Basel)*, vol. 12, no. 3148, Jul. 2023, doi: 10.3390/electronics12143148.

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

[9] D. Maiorca and B. Biggio, "Digital Investigation of PDF Files: Unveiling Traces of Embedded Malware," *IEEE Secur Priv*, vol. 17, no. 1, pp. 63–71, Jan. 2019, doi: 10.1109/MSEC.2018.2875879.

[10] N. Fleury, T. Dubrunquez, and I. Alouani, "PDF-Malware: An Overview on Threats, Detection and Evasion Attacks," 2021.

[11] A. M. Barmawi and D. Pradeka, "Information hiding based on histogram and pixel pattern," *Journal of Cyber Security and Mobility*, vol. 6, no. 4, pp. 397–425, Oct. 2017, doi: 10.13052/jcsm2245-1439.642.

[12] D. Pradeka, "Penyembunyian Informasi dengan Metode Crypto-Steganography menggunakan Media Gambar Berbasis Mobile," *Sistem Informasi Manajemen dan Keuangan dalam Industri 4.0*, pp. 104–111, 2018.

[13] D. Pradeka, "Implementasi Aplikasi Kriptografi Berbasis Android menggunakan Metode Subtitusi dan Permutasi," *In Search – Informatic, Science, Entrepreneur, Applied Art, Research, Humanism*, vol. 18, no. 01, pp. 161–168, Apr. 2019.

[14] S. S. Pachpute, "Malware Analysis on PDF," San Jose State University, San Jose, CA, USA, 2019. doi: 10.31979/etd.pf8d-htjh.

[15] D. Stevens, "PDF Tools," Didier Stevens. Accessed: Dec. 25, 2023. [Online]. Available: https://blog.didierstevens.com/programs/pdf-tools/

[16] P. Singh, S. Tapaswi, and S. Gupta, "Malware Detection in PDF and Office Documents: A survey," *Information Security Journal*, vol. 29, no. 3, pp. 134–153, May 2020, doi: 10.1080/19393555.2020.1723747.

[17] M. Issakhani, P. Victor, A. Tekeoglu, and A. H. Lashkari, "PDF Malware Detection based on Stacking Learning," in *International Conference on Information Systems Security and Privacy*, Science and Technology Publications, Lda, 2022, pp. 562–570. doi: 10.5220/0010908400003120.

[18] W. Suteddy, D. Aprianti, R. Agustini, A. Adiwilaga, and A. Atmanto, "End-To-End Evaluation of Deep Learning Architectures for Offline Handwriting Writer Identification: A Comparative Study," *JOIV : Int. J. Inform. Visualization*, vol. 7, no. 1, p. 178185, Mar. 2023.

[19] A. N. Syafia, M. F. Hidayattullah, and W. Suteddy, "Studi Komparasi Algoritma SVM dan Random Forest pada Analisis Sentimen Komentar Youtube BTS," *Jurnal Informatika: Jurnal pengembangan IT (JPIT)*, vol. 8, no. 3, pp. 207–212, Sep. 2023.

[20] R. Fettaya and Y. Mansour, "Detecting malicious PDF using CNN," Jul. 2020.

[21] A. Géron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed. Sebastopol: O'Reilly, 2019.

[22] Amita. Kapoor, *Hands-On Artificial Intelligence For IoT*. PACKT Publishing Limited, 2019.

[23] A. Rahmah, N. Sepriyanti, M. H. Zikri, I. Ambarani, and M. Yusuf Bin Shahar, "Implementation of Support Vector Machine and Random Forest for Heart Failure Disease Classification," *Public Research Journal of Engineering, Data Technology and Computer Science*, vol. 1, no. 1, pp. 34–40, Jul. 2023.

[24] R. I. Arumnisaa and A. W. Wijayanto, "Perbandingan Metode Ensemble Learning: Random Forest, Support Vector Machine, AdaBoost pada Klasifikasi Indeks Pembangunan Manusia (IPM)," *Januari*, vol. 12, no. 1, pp. 206–2018, 2023.

[25] M. Wainberg, B. Alipanahi, and B. J. Frey, "Are Random Forests Truly the Best Classifiers?," 2016.

[26] S. A. Roseline, S. Geetha, S. Kadry, and Y. Nam, "Intelligent Vision-Based Malware Detection and Classification Using Deep Random Forest Paradigm," *IEEE Access*, vol. 8, pp. 206303–206324, 2020, doi: 10.1109/ACCESS.2020.3036491.

[27] H. Pramoedyo, D. Ariyanto, and N. N. Aini, "Comparison of Random Forest and Naïve Bayes Methods for Classifying and Forecasting Soil Texture In The Area Around Das Kalikonto, East Java," *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, vol. 16, no. 4, pp. 1411–1422, Dec. 2022, doi: 10.30598/barekengvol16iss4pp1411-1422.

[28] D. Pradeka, A. Adiwilaga, D. A. R. Agustini, M. B. Hidayatullah, and A. Suheryadi, *Belajar Dasar Pemrograman Web serta Pengenalan Kriptografi dan Plugin Moodle*, vol. 1. Bandung: Widina Media Utama, 2023.

[29] N. Nofriani, "Machine Learning Application for Classification Prediction of Household's Welfare Status," *JITCE (Journal of Information Technology and Computer Engineering)*, vol. 4, no. 02, pp. 72–82, Sep. 2020, doi: 10.25077/jitce.4.02.72-82.2020.

[30] D. Avelino, L. Cancerlon, M. K. Ryanta, Y. H. Christianto, and W. Wangnardy, "Penggunaan Bahasa Pemrograman Python dalam Menganalisis Perbedaan Desain Website Tren di Negara Jepang dan Dunia," *Journal of Student Development Information System (JoSDIS)*, vol. 3, no. 2, pp. 51–61, 2023.

[31] tpn, "Technically-oriented PDF Collection," Github. Accessed: Dec. 26, 2023. [Online]. Available: https://github.com/tpn/pdfs

[32] ahlashkari, "PDFMalLyzer," Behavior-Centric Cybersecurity Center (BCCC).

[33] J. X. McKie, "PyMuPDF," Artifex Software, Inc. Accessed: Dec. 26, 2023. [Online]. Available: https://pymupdf.io

[34] A. Althnian *et al.*, "Impact of dataset size on classification performance: An empirical evaluation in the medical domain," *Applied Sciences (Switzerland)*, vol. 11, no. 2, pp. 1–18, Jan. 2021, doi: 10.3390/app11020796.

[35] F. Baharuddin and A. Tjahyanto, "Peningkatan Performa Klasifikasi Machine Learning Melalui Perbandingan Metode Machine Learning dan Peningkatan Dataset," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 11, no. 1, pp. 25–31, Mar. 2022, doi: 10.32736/sisfokom.v11i1.1337.

[36] R. Imantiyar, ; Dhomas, and H. Fudholi, "Kajian Pengaruh Dataset dan Bias Dataset terhadap Performa Akurasi Deteksi Objek," *PETIR: Jurnal Pengkajian dan Penerapan Teknik Informatika*, vol. 14, no. 2, 2021, doi: 10.33322/petir.v14i2.1150.

[37] K. Koptyra and M. R. Ogiela, "Distributed steganography in PDF files - Secrets hidden in modified pages," *Entropy*, vol. 22, no. 6, Jun. 2020, doi: 10.3390/E22060600.