Journal Homepage: -www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI:10.21474/IJAR01/21758
DOI URL: http://dx.doi.org/10.21474/IJAR01/21758

RESEARCH ARTICLE

ENHANCING PDF MALWARE CLASSIFICATION USING CTGAN-BASED DATA AUGMENTATION AND SUPERVISED LEARNING

Amadou Diabagate[1], Yazid Hambally Yacouba[2], Adama Coulibaly[1] and Abdellah Azmani[3]

1. Faculty of Mathematics and Computer Science, University Felix Houphouët-Boigny, Cote d'Ivoire.
2. National High School of Architecture and Urban Planning, University of Bondoukou, Cote d'ivoire.
3. Faculty of Sciences and Technologies, University AbdelmalekEssaadi, Tangier, Morocco.

Manuscript Info

Manuscript History
Received: 11 July 2025
Final Accepted: 13 August 2025
Published: September 2025

Key words: -
Malware detection, PDF security, Artificial intelligence, CTGAN-based data augmentation, Machine learning, Cyber threat modeling, Smart cybersecurity solutions

Abstract

The increasing sophistication of cyberattacks exploiting PDF files poses a critical challenge to digital security. This study presents an intelligent detection framework that combines synthetic data augmentation and cutting-edge machine learning techniques to identify malicious PDF documents with high precision. To address the issue of class imbalance often found in cybersecurity datasets, we employ Conditional Tabular GAN (CTGAN) to generate realistic synthetic samples, thereby enriching the training set and improving the generalization capability of classifiers.Six supervised models are assessed, Decision Tree, Random Forest, XGBoost, Support Vector Machine, Naive Bayes, and Neural Network, using the augmented dataset. Among them, XGBoost consistently delivers the most robust performance. To foster transparency and trust, the framework integrates SHapley Additive exPlanations (SHAP), enabling a clear interpretation of feature contributions to each classification decision.Overall, this work introduces a comprehensive and explainable approach to strengthening PDF document security, offering a promising path for deployment in sensitive organizational environments such as government, education, and enterprise systems.

Introduction: -

Information systems have become indispensable to the functioning of modern organizations, underpinning operations in sectors as diverse as education, finance, insurance, industry, and national security [1]. Far beyond their role in managing data storage and transfer, these systems support communication, ensure operational continuity, and guide strategic decisions in increasingly complex digital environments.

Yet, this dependence on digital infrastructures has also brought about new forms of vulnerability. The value of the data housed in these systems makes them prime targets for cybercriminals exploiting weaknesses in their architecture [2]. As cyber threats grow in frequency, sophistication, and stealth, protecting these systems has become a strategic imperative [3].

Corresponding Author: -Amadou Diabagate
Address: -Faculty of Mathematics and Computer Science, University Felix Houphouet-Boigny, Cote dIvoire.

In response, the field of cybersecurity has turned increasingly toward artificial intelligence (AI) as a means of bolstering detection and prevention efforts [4]. Machine learning, deep learning, and fuzzy logic have shown considerable promise in analyzing large volumes of data in near real time, identifying anomalous behaviors, and even anticipating intrusions before they occur [5], [6]. This convergence between AI and cybersecurity has paved the way for more adaptive, autonomous, and resilient digital defense systems [7].

Within this dynamic landscape, AI-enhanced systems can continuously evolve alongside emerging threats, offering faster incident responses and more intelligent risk mitigation [8]. Such capabilities are essential to safeguarding sensitive digital assets and maintaining trust in critical infrastructures.

This paper focuses on one particularly vulnerable yet ubiquitous element of digital communication: the PDF file. Despite its widespread use for document exchange, the PDF format is frequently exploited to conceal and deliver malicious code [9].

To tackle the growing threat of malicious PDF files, we present an artificial intelligence driven framework that combines two powerful yet often separately applied techniques: supervised machine learning and generative data augmentation using Conditional Tabular GAN (CTGAN). One of the key challenges in this domain is the imbalance between benign and malicious instances, with our dataset containing approximately 32 percent malicious files versus 68 percent benign ones. To address this disparity and enhance the quality of the training set, we leveraged CTGAN to generate synthetic benign samples that preserve the statistical structure of the original data [10].

What sets our approach apart is this thoughtful integration of generative modeling and predictive learning. CTGAN enriches the dataset with realistic examples that help mitigate bias and improve generalization, while supervised machine learning algorithms, optimized through systematic hyperparameter tuning, deliver accurate and consistent classification results. This combined strategy offers a practical and effective contribution to the ongoing effort to secure PDF documents against evolving cyber threats [11].

Existing PDF-malware detectors often train on imbalanced corpora with minimal resampling, use GAN-based augmentation without rigorous statistical checks (sometimes beyond the training split), and depend on deep models that are difficult to audit in SOC settings. We propose an integrated pipeline that keeps augmentation train-only, validates CTGAN samples with Kolmogorov-Smirnov tests, correlation preservation, and PCA overlap, couples them with a transparent boosted classifier (XGBoost), and provides SHAP explanations at both global and per-instance levels. This combination remains uncommon and directly supports traceable decisions and rapid triage in production.

The remainder of this paper is structured as follows. Section 2 outlines the security vulnerabilities inherent in PDF files and the rationale for automated detection. Section 3 reviews relevant literature on machine learning and data augmentation in cybersecurity. Section 4 presents the methodological framework. Sections 5 and 6 present the data augmentation process using CTGAN and the supervised classification approach employed to detect malicious PDF files. Section 7 discusses key findings and limitations, and Section 8 concludes with perspectives for future research.

PDF FILE SECURITY:-

Information system security has become a cornerstone of modern digital infrastructures, playing a critical role in ensuring operational continuity and protecting sensitive data assets. Commonly referred to as cybersecurity, this domain focuses on safeguarding information systems, networks, software, and data from attacks, unauthorized access, and corruption. Its fundamental objectives are to preserve data confidentiality, integrity, and availability throughout storage, processing, and transmission phases [12].

File security forms an integral part of effective cybersecurity frameworks [5]. It encompasses the protection of digital files from unauthorized access, illegal modification, deletion, damage, and degradation. Such protective measures are especially crucial for documents containing sensitive personal data, proprietary corporate information, or classified governmental content.

PDF file security has emerged as a particularly pressing concern, given the widespread use of the Portable Document Format for disseminating confidential materials such as contracts, financial statements, and personal records. Despite their convenience, PDF files are inherently vulnerable to various forms of exploitation, including

unauthorized access, malicious tampering, data exfiltration, and embedded code injection [9], [13]. Among these threats, unauthorized access remains one of the most prevalent, especially when documents lack password protection or robust encryption. In addition, embedded metadata, such as author information, creation timestamps, or reviewer comments, can inadvertently disclose sensitive content if not properly sanitized [6].

To counteract these risks, a variety of protective strategies have been developed. These include encryption, digital signatures, malicious script detection, metadata removal, and watermarking [14].Encryption remains one of the most effective means of securing PDF files against unauthorized access. It can be configured to restrict both file opening (via an open password) and specific operations such as printing or editing (via a permission password). Digital signatures also play a pivotal role in ensuring document authenticity and integrity. By verifying that a file has not been altered post-creation and confirming the author's identity, digital signatures are particularly critical in legal and financial contexts [9].

Moreover, PDF files can contain embedded JavaScript code, which may serve as a vector for exploiting vulnerabilities in PDF readers. As such, script detection and removal are crucial [15]. Metadata cleansing is equally important, particularly when documents are disseminated publicly, to protect user anonymity and confidentiality [12]. Lastly, watermarking techniques, including visible and invisible digital watermarks, serve to deter unauthorized copying and support provenance tracking [16].Recent incident reports repeatedly highlight three recurring vectors in PDF-borne attacks: embedded JavaScript, abuse of launch actions, and manipulation of cross-reference (xref) tables. These patterns justify automated, large-scale screening. Our feature design targets exactly these behaviorsso that model signals align with known exploitation tactics and remain easy for analysts to interpret[9], [13].

With the growing adoption of cloud-based services, securing PDF files in distributed environments has become even more challenging. To address this, advanced schemes combining encryption and key management have been proposed, enabling access control and data protection in shared, multi-user ecosystems [17].

RELATED WORK: -

Over the past decade, the detection of malicious PDF files has drawn growing attention in the field of cybersecurity, driven by the increasing sophistication of attacks and the widespread use of the PDF format as a vector for embedded threats. Early approaches to PDF malware analysis largely relied on static inspection techniques, focusing on structural attributes or the presence of embedded scripts. For instance, Laskov et al. [13] demonstrated that analyzing JavaScript code within PDF files could reveal significant indicators of compromise. Bayer et al. [15] introduced TTAnalyze, a dynamic analysis tool designed to simulate file execution in sandboxed environments, laying the groundwork for behavior-based detection models.

As attacks became more evasive and polymorphic, researchers began turning to machine learning as a more adaptive solution. Smutz and Stavrou [18] showed how classifiers like Support Vector Machines and Decision Trees could leverage metadata and structural features to detect malicious content. Later work by Raff et al. [19] employed convolutional neural networks to analyze binary representations of documents, while Wang et al. [20] explored recurrent neural networks with attention mechanisms to capture temporal dependencies in embedded scripts. These efforts significantly improved detection rates, especially in scenarios where traditional rule-based methods failed.

Despite these advancements, one persistent challenge remains: class imbalance. In real-world datasets, malicious samples are often vastly outnumbered by benign files, leading to biased models and poor generalization. To address this, recent research has explored the use of generative models for data augmentation. Conditional Tabular GAN (CTGAN), introduced by Xu et al. [21], represents a significant innovation in this area, offering the ability to synthesize high-quality tabular data with complex interdependencies. Several studies have validated the use of GAN-based augmentation to improve classifier performance, particularly in imbalanced learning contexts [22], [23].

While most prior work treats data generation and classification as independent components, our approach seeks to integrate them into a cohesive and streamlined pipeline. Synthetic data produced by CTGAN is used not only to rebalance the training set but also to improve the diversity and representativeness of the input space. This enriched dataset is then processed using a diverse array of supervised learning models including Decision Tree, Support Vector Machine, Naive Bayes, Neural Network, Random Forest, and XGBoost.

In contrast to earlier frameworks that often relied on opaque deep learning architectures or rigid feature engineering pipelines, our approach adopts a more transparent and versatile strategy. By combining synthetic data generation through CTGAN with a broad range of supervised learning models, we offer a methodology that is both interpretable and empirically robust. This allows us to achieve high performance while maintaining clarity in the decision-making process.

METHODOLOGY:-

This study introduces a twofold methodological framework that combines generative data augmentation with supervised machine learnig techniques to improve the detection of malicious PDF files. The overall process rests on two key pillars. The first is the use of Conditional Tabular GAN (CTGAN) to address data imbalance by generating high-quality synthetic samples [21]. The second is the application of a diverse set of classification algorithms to evaluate the impact of the enriched dataset on detection performance [11]. Figure 1 below provides an overview of the system's architecture, illustrating the overall workflow.

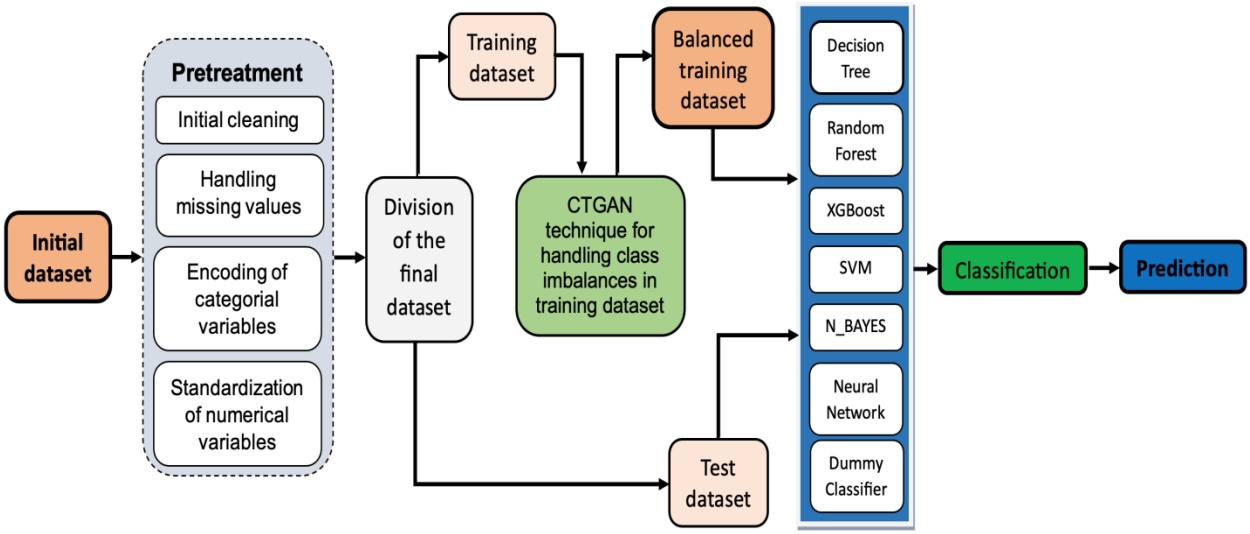

Figure 1: - Overview of the machine learning pipeline used

To operationalize this workflow, the pseudocode below details each technical stage used to build, validate, and explain the predictive model.

Algorithm 1 – Machine Learning Workflow for Malicious PDF Detection
Input:
Structured dataset on PDF files
Output:
PDF files classification and interpretability insights

Step 1: Split dataset into training (80%) and testing (20%)
Step 2: The training set was augmented with CTGAN, ensuring equal class distribution (test set unchanged).
Step 3: Model suite:Decision Tree, Random Forest, XGBoost, SVM, Naïve Bayes, Neural Network; Dummy as non-informative baseline
Step 4: Hyperparameter optimization viaRandomizedSearchCV (K-fold stratified cross-validation)
Step 5: Model selection using key metrics(Accuracy, F1 score, Precision, Recall, AUC-ROC, MCC)
Step 6: Post-hoc explanations with SHAP (global summaries and per-instance attributions).

Dataset Overview and Preparation :
The dataset employed in this study comprises 6,343 PDF files, split into 4,315 benign and2,028 malicious instances. Each file is described by 22 static features extracted through a lightweight static analysis process. These features are designed to capture both general characteristics (such as file size, metadata volume, number of pages) and structural

properties (such as the presence of JavaScript code, embedded objects, or suspicious triggers). A detailed summary of the features is provided in Table 1**.**

Table 1: - General and Structural Characteristics of the PDF Files

FEATURE NAME	DESCRIPTION
General Characteristics	
Total size of PDF	File size in bytes
Length of title text	Legitimate PDFs typically have meaningful, descriptive titles
Encryption enabled	Indicates whether the file is password-protected
Volume of metadata	Size of the descriptive information embedded in the PDF
Number of pages	Total number of pages in the document
Header existence	Indicates the PDF version used
Number of Embedded Images	Total number of images embedded in the document
Text Presence	Indicates whether the PDF contains readable text
Object Count	Total number of objects (text, images, streams, fonts, annotations, etc.)
Embedded Files	Number of additional files attached to the PDF
Structural Characteristics	
Stream object count	Number of binary data streams within the PDF
JavaScript presence	Number of objects containing JavaScript code
Automatic Action	Defines specific actions triggered by events (often used with malicious JS)
Launch command	Executes commands/programs (often used for data theft or malware)
Open trigger	Specifies actions upon opening, often tied to malicious JavaScript
AcroForm Tag Count	Acrobat forms potentially containing exploitable scriptable fields
JBig2Decode Filter Presence	Indicates use of JBig2Decode, often used to encode malicious content
Xref Length	Number of cross-reference tables managing the object structure
XFA Form Used	Indicates presence of XML-based forms supporting scripting
Xref Entry Count	Number of entries in the Xref tables, often malformed in malicious PDFs
Rich media presence	Number of embedded multimedia or Flash objects
Trailer tag count	Number of trailer sections; abnormal counts may indicate suspicious content

Before any modeling or data generation, a standard preprocessing pipeline was applied to ensure consistency and quality across the dataset. Missing values were imputed using the most frequent value for each feature via the SimpleImputer method from Scikit-learn. Categorical features were label-encoded, while numerical features were standardized using StandardScaler, improving convergence for subsequent training phases. Finally, the dataset was split into training (80%) and testing (20%) subsets, preserving the original class distribution through stratified sampling.

Descriptive Statistical Analysis:

To better understand the internal characteristics of the dataset, a descriptive statistical analysis was carried out on the numerical and categorical variables extracted from the PDF files. This preliminary exploration reveals significant structural variability and several features that may play a decisive role in the classification process. Table 2 presents a summary of the key numerical variables, including their average values, variability, and extreme observations. These figures help highlight both the general trends and the degree of dispersion within the dataset.

Table 2: –Descriptive statistics of numerical variables

VARIABLE	MEAN	VARIANCE	STANDARD DEVIATION	MEDIAN	MODE	RANGE
Total size of PDF	95,95	200460,54	447,73	60	9	23817
Volume of metadata	317,08	911116,98	954,52	283	180	50284
Number of pages	4,07	94,78	9,74	1	1	201
xref Length	1816,07	206465416,84	14368,90	32	21	263988
Length of title text	24,33	510354,68	714,39	0	0	50093
Count of embedded objects	-0,01	0,04	0,19	0	0	6
Stream object count	24,06	1727,18	41,56	10	2	813
Trailer tag count	1,35	1,75	1,32	1	1	47
Object stream count	2,15	29,89	5,47	0	0	69
JavaScript presence	0,69	33,73	5,81	0	0	405
AcroForm usage	0,44	0,61	0,78	0	0	7
Automatic actions	0,40	46,71	6,83	0	0	214
Object tag count	66,81	26706,97	163,42	29	9	7077
XFA form used	-0,01	0,07	0,27	0	0	6
JBIG2 usage	0,03	0,45	0,67	0	0	15
Image presence	3,02	195,36	13,98	0	0	593

The total size of PDF files shows a mean of 95.95 KB, but this average conceals a considerable dispersion, as indicated by a standard deviation of 447.73 KB and a maximum value reaching 23,816 KB. The volume of metadata is another highly variable attribute, with a mean of 317.08 KB and a variance exceeding 900,000, pointing to extreme disparities among the documents. A similar pattern is observed in the xref length variable, which reaches values up to 263,987 units, far above the median of 32. This strong asymmetry suggests the presence of outliers or highly complex internal structures in certain files.

Some features, although sparse, may carry important security signals. Variables such as JavaScript presence, automatic actions, and AcroForm usage show low average values but unusually high standard deviations. This means that while the majority of documents do not contain these elements, a small subset does so in a disproportionately large way, which may be relevant for detecting malicious behaviors.Other indicators such as the count of embedded objects, XFA form usage, and JBIG2 compression appear very rarely in the dataset. Their mean values are close to zero, but their occasional presence might still be meaningful in a security context.

The analysis of categorical variables adds further perspective. The dataset is composed of 68.03% benign documents and 31.97% malicious ones, with the benign class clearly dominating. Regarding the presence of text, the distribution is almost balanced, with 50.54% of documents containing visible text, 46.32% lacking it, and 3.14% marked as unclear. The launch command, often associated with exploitation mechanisms, appears in only 3.77% of the files, confirming its rarity but also its potential risk when it is present.

In summary, this descriptive analysis reveals a dataset marked by high heterogeneity, numerous outliers, and a mix of common and rare behaviors. These insights are crucial for guiding the next steps of the analysis, particularly in relation to feature engineering, normalization, and robust model design.

Generative Data Augmentation Using CTGAN:
To improve the balance and diversity of the dataset, and to strengthen the learning capacity of our classification models, we incorporated a generative data augmentation phase using CTGAN. The next two subsections explain the motivations behind this choice, along with the principles and mechanisms that guided its implementation.

Motivation and Implementation Strategy:
An important challenge in the development of malware detection systems lies in the inherent imbalance between benign and malicious files. In our dataset, this disparity was clearly visible, with 4,315 benign PDFs compared to only 2,028 malicious ones. Such imbalance often causes machine learning models to favor the majority class, which can lead to misleading accuracy scores and a dangerous drop in sensitivity to rare but critical threats. This limitation is particularly problematic in cybersecurity, where undetected attacks may have severe consequences. To address this issue, we adopted the CTGAN model (Conditional Tabular GAN) to generate synthetic benign samples.

Rather than using traditional oversampling methods, CTGAN enabled us to create realistic data points that preserved the statistical structure of the original dataset. This step was essential to reduce class imbalance and ensure that the learning process remained fair and stable. It also contributed to better generalization, especially in complex classification scenarios where the boundaries between benign and malicious behavior are subtle. The choice of CTGAN was guided by the nature of our data. Our features were extracted from static analysis of PDF files and organized in tabular form, which aligns well with CTGAN's design.

This model has already shown promising results in similar contexts, particularly when both numerical and categorical variables must be modeled jointly and coherently [21], [22]. By incorporating CTGAN into the data preparation workflow, we aimed to build a more balanced and resilient training set. This generative approach strengthened the overall reliability of the pipeline, providing a solid foundation for the supervised learning methods applied in the next stages of the study.

Mathematical Foundations of CTGAN:
CTGAN (Conditional Tabular GAN) is an adaptation of Generative Adversarial Networks (GANs) for tabular data containing both numerical and categorical variables. It relies on an adversarial architecture involving two neural networks: a generator G and a discriminator D, which compete in a zero-sum game [21].

- **General Formulation of GANs:**

In a standard GAN, the generator $G(\mathcal{Z})$ produces synthetic data from a random noise $\mathcal{Z} \sim p_{\mathcal{Z}}(z)$, while the discriminator D(x) attempts to distinguish real data $x \sim p_{data}(x)$ from fake data $G(\mathcal{Z})$. The objective function is given by:

$$\min_G \max_D V(D,G = \mathbb{E}_{x\sim pdata}[\log D(x)] + \mathbb{E}_{\sim p_{\mathcal{Z}}}[\log(1 - D(G(\mathcal{Z})))] (1)$$

This leads to a minimax game, where G learns to fool D, and D learns to resist being fooled.

- **CTGAN Specificities:**

CTGAN enhances GANs for tabular data through two mechanisms. The generator is conditioned on categorical variables to capture dependencies between discrete and continuous features:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim pdata}[\log D(x)] + \mathbb{E}_{\sim p_{\mathcal{Z},c\sim p_c}}[\log(1 - D(G(\mathcal{Z};c),c))] (2)$$

Numerical variables are modeled with Gaussian Mixture Models (GMMs) to preserve realistic continuous distributions:

$$M(x) = \sum_{k=1}^{K} \pi_k N(\mu_{k,\sigma_k}) \quad (3)$$

Application of Machine Learning:
Our task relies on tabular, static features extracted from PDFs (22 variables). In this setting, tree-based ensembles consistently perform well while remaining efficient and interpretable, as shown in prior work on PDF malware and malware detection more broadly [6], [8], [24]. We therefore emphasize a diverse set of non-deep learners (Decision Tree, Random Forest, XGBoost, SVM, Naive Bayes) and keep a shallow MLP only as a lightweight deep baseline. Heavier deep architectures were intentionally excluded given the moderate data size and the sensitivity of deep models to class imbalance [11], [25], as well as the operational efficiency and explainability required in SOC environments [5].

To complement this modeling strategy, we also incorporated a simple baseline classifier (DummyClassifier), which serves as a minimal benchmark for performance evaluation. This model, configured with the most_frequent strategy, always predicts the majority class from the training set. While not intended for practical deployment, it provides a useful reference point for assessing how much the actual classifiers improve over naive or random predictions.

Description of the Selected Machine Learning Algorithms:
In the paragraphs that follow, we provide a concise overview of the six machine learning models applied in this study, highlighting their underlying mechanisms, relevance to PDF threat detection tasks, and known limitations when deployed in real-world cybersecurity environments.

Decision Tree algorithms are among the most interpretable classification methods. They recursively partition the dataset based on impurity measures such as Gini index or entropy, resulting in a tree structure that offers transparency and ease of interpretation [26]. However, decision trees are prone to overfitting, particularly with small or noisy datasets, unless regularized by limiting tree depth or pruning.

Random Forest**,** introduced by Breiman, extends the capabilities of decision trees by aggregating predictions from multiple trees trained on random subsets of data [27]. This ensemble approach enhances generalization and reduces model variance [28]. It also provides robust estimates of feature importance, which is essential in security applications that require traceability and decision auditability.

XGBoost (eXtreme Gradient Boosting) is a high-performance boosting algorithm developed by Chen and Guestrin [29]. It incorporates regularization, parallel training, and pruning mechanisms to achieve outstanding predictive accuracy. XGBoost is particularly suited for imbalanced and noisy datasets, which are common in real-world malware detection scenarios [29], [30].

Support Vector Machines (SVMs) aim to find the optimal hyperplane that separates classes with maximum margin [31]. While effective for linearly separable data, their performance tends to degrade with noisy or overlapping distributions, which are frequently encountered in complex PDF classification problems [24].

Naive Bayes is a simple yet effective probabilistic classifier that applies Bayes' theorem under the assumption of conditional independence between features. While this assumption is rarely fully met in practice, the algorithm has proven robust across a variety of real-world tasks, especially when working with high-dimensional or sparse data. In the context of malware detection, Naive Bayes provides fast training and inference, making it attractive for scenarios requiring quick decision-making. However, its performance tends to decline when feature correlations are strong or when the dataset is heavily imbalanced [32], [33].

Multilayer Perceptron (MLP) represents a type of feedforward neural network composed of multiple layers of interconnected neurons [34], [35]. Capable of capturing nonlinear patterns in data, MLPs are powerful classifiers but require large, well-balanced datasets for stable training. Their limited interpretability can also hinder their adoption in security-critical systems, where explainability is crucial [36].

As a point of reference, we added a very simple baseline model using Scikit-learn'sDummyClassifier, configured with the most_frequent strategy. This model, while clearly not suitable for any practical application, simply predicts the most common class in the training data. Its purpose is not performance, but perspectiveit helps illustrate what minimal predictive ability looks like, making it easier to appreciate the real contributions of the more sophisticated algorithms tested in this study [37], [38].

In summary, the comparative evaluation of these models provides a balanced perspective on their predictive accuracy, interpretability, and practical applicability for detecting malicious PDF files within a cybersecurity framework.

Hyperparameter Optimization and Model Selection:

The dataset used in this study comprises 6,343 PDF files, including 4,315 benign and 2,028 malicious instances, and features a significant class imbalance. Each file is described by 22 static features, categorized into general and structural characteristics, which reflect inherent file-level properties relevant to security analysis.

To improve classification performance, six supervised machine learning algorithms were subjected to systematic hyperparameter tuning using the RandomizedSearchCV method. This approach enables probabilistic sampling of the hyperparameter space, offering a more computationally efficient alternative to exhaustive techniques such as GridSearchCV [39],[40].Table 3 presents an overview of the selected algorithms alongside the optimal hyperparameter settings that achieved the highest predictive performance on the imbalanced dataset.

Table 3:-Summary of Machine Learning Models and their Optimized Hyperparameter Settings

Model	Optimized Hyperparameters and Values
Decision Tree	criterion = entropy; max_depth = None; min_samples_leaf = 1; min_samples_split = 4
Random Forest	bootstrap = False; max_depth = None; max_features = log2; min_samples_leaf = 2; min_samples_split = 13; n_estimators = 129
XGBoost	colsample_bytree= 0.5477050582452057; gamma = 0.18540912609913318; learning_rate= 0.2106523757990822; max_depth = 7; n_estimators = 198; subsample = 0.7956488938538635
SVM	C = 9.74755518841459; gamma = scale; kernel = rbf
Naive Bayes	var_smoothing= 0.01873817422860387
Neural Network	Activation = tanh; alpha =0.0022233911067827614; early_stopping = True; hidden_layer_sizes = (100; 50); learning_rate = adaptive; n_iter_no_change = 10; solver = adam; validation_fraction = 0.1

Hyperparameter tuning was performed using cross-validation to ensure that the selected configurations maximize the models' generalization capacity. This step is critical for ensuring the reproducibility of the results and establishing a robust foundation for the comparative analysis of model performance. To rigorously assess model performance, we implemented a Stratified K-Fold cross-validation strategy.

This method partitions the dataset into K subsets (folds) while preserving the class distribution across each fold, a particularly important consideration in imbalanced classification tasks. During each iteration, one fold is used for testing, while the remaining K–1 folds serve as the training set. This process is repeated Ktimes, and the results are averaged to provide a more robust estimate of model performance. In our case, we set K to 10, a commonly recommended choice in the literature [41], [42], which offers a good trade-off between bias and variance. This evaluation protocol plays a central role in ensuring fairness and consistency across models when comparing their predictive capabilities.

Performance Metrics of Machine Learning Models:

In classification tasks, evaluating the effectiveness of predictive models relies on four fundamental outcomes: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). These values serve as the foundation for key performance metrics used to assess the robustness and reliability of each algorithm. Below, we outline the mathematical definitions of some of the most widely used evaluation metrics applied in this study.

Accuracy, or overall correctness, reflects the proportion of correct predictions, both positive and negative, over the total number of instances. It provides a general measure of how often the model makes correct predictions:

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

Recall (or True Positive Rate) evaluates the model's ability to correctly identify malicious files, which is essential in security applications where missing threats could have serious consequences:

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

Precision quantifies the proportion of correctly predicted positive cases among all cases predicted as positive. It reflects the reliability of positive predictions made by the model.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (6)$$

F1-score represents the harmonic mean between Precision and Recall. It offers a balanced metric that is especially useful in datasets with class imbalance, as it accounts for both false positives and false negatives.

$$\text{F1} - \text{s}$$

Matthews Correlation Coefficient (MCC) provides a balanced measure that accounts for all four outcomes (TP, TN, FP, FN), and is particularly effective when the classes are imbalanced. It returns a value between -1 and +1, where +1 indicates perfect prediction, 0 no better than random, and -1 total disagreement between prediction and observation:

$$\text{MCC} = \frac{(\text{TP} \times \text{TN}) - (\text{FP} \times \text{FN})}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \quad (8)$$

Area Under the ROC Curve (AUC-ROC**)** quantifies the model's ability to distinguish between classes across all classification thresholds. It plots the true positive rate (Recall) against the false positive rate (FPR), where:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (9)$$

Together, these metrics provide a comprehensive view of model performance, supporting fair comparison and selection of the most effective classification approach for malicious PDF detection.

Data Augmentation with CTGAN: -

To address the class imbalance inherent in the dataset and to enhance the generalization ability of the predictive models, we incorporated a generative augmentation strategy based on Conditional Tabular GANs (CTGAN). This approach involves generating synthetic data samples that are statistically consistent with the real dataset, thereby improving training diversity and reducing overfitting risks.

Before being included in the training pipeline, the quality of these synthetic samples was thoroughly assessed to ensure both their statistical validity and structural fidelity [21]. This section presents the validation methodology and results, focusing on the marginal distributions of key features and their joint statistical behavior compared to the empirical data.

Univariate Distribution Analysis and Statistical Alignment:

To assess the fidelity of the synthetic samples produced by CTGAN, a comparative analysis was carried out on six critical features, selected for their relevance in the classification process and their diverse typological nature, spanning structural, behavioral, and content-based dimensions:

- JavaScript presence (hasJS) – a binary indicator of embedded scripting.
- Volume of metadataandTotal size of PDF – continuous proxies of document complexity and density.
- Open trigger (hasOpenAction)andAutomatic actions (hasLaunchAction) – behavioral flags reflecting documentinteractivity.
- xref Length (xrefLength) – a discrete structural metric indicative of internal referencing.

The comparative plots are presented in Figure 2, where the synthetic distributions generally track closely with their realcounterparts, particularly for continuous variables like metadata volume and file size, where the density curves are almost superimposed. This visual consistency suggests that CTGAN successfully modeled the probabilistic structures underlying the original data.

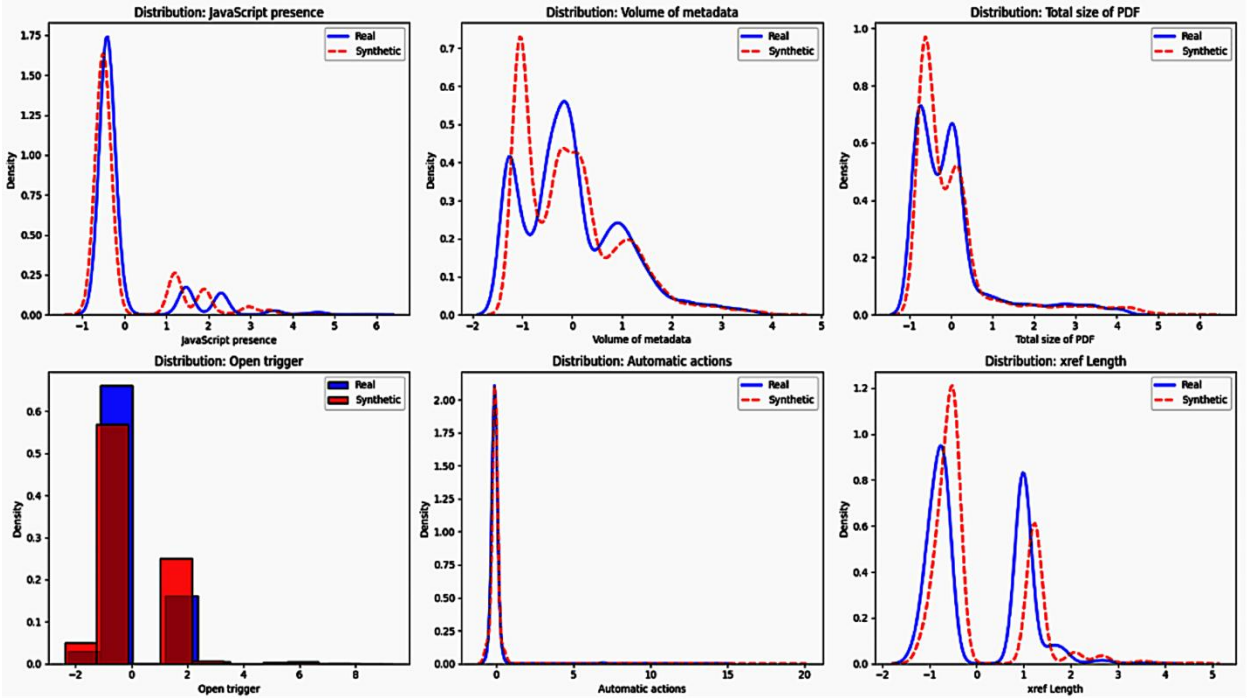

Figure 2: -Comparative Distributions of Real and Synthetic Data for Key Predictive Features

To strengthen these observations, the Kolmogorov–Smirnov (KS) testwas applied to each variable. The results, summarized in Table 4**,** show that for three of the six features (JavaScript presence, Open trigger, and Launch actions), the null hypothesis of identical distributions cannot be rejected ($p > 0.05$). Although slight discrepancies are noted for continuous and discrete features such as volume of metadata, total size of PDF, and xref length, these variations remain within tolerable limits for augmentation purposes and do not introduce systematic bias [43].
Despite minor deviations, the overall distributional structure is sufficiently preserved to ensure data representativeness, thereby reinforcing classifier performance while addressing imbalance-related vulnerabilities.

Table 4: -Kolmogorov–Smirnov test comparing distributions between real and synthetic data for six key features

Variable	**Type**	**KS Statistic**	**p-value**	**Distribution Equal(p > 0.05)**
Javascript presence	Binary	0.0	1.0	True
Volume of metadata	Continuous	0.1689	0.0	False
Total size of PDF	Continuous	0.1914	0.0	False
Open trigger	Binary	0.0	1.0	True
Launch	Binary	0.0	1.0	True
Xref Length	Discrete	0.3197	0.0	False

Collectively, these results affirm that the synthetic data generated by CTGAN maintains a high degree of statistical realism, validating its integration into the model training pipeline.

Multivariate Coherence and Structural Integrity:

Beyond individual feature alignment, the structural validity of the synthetic data was assessed through multivariate analysis techniques to determine whether CTGAN preserved the complex interdependencies among variables.

A first level of evaluation involved Pearson correlation matrices, independently computed for both real and synthetic datasets. As illustrated in Figure 3**,** the matrices exhibit a high level of congruence, particularly for feature pairs involving `size`, volume of metadata, and `xref_length`, which consistently show strong positive correlations. The difference in correlation coefficients between real and synthetic data remained within a ±0.05 margin, indicating that the generative model captured the essential joint relationships without distortion.

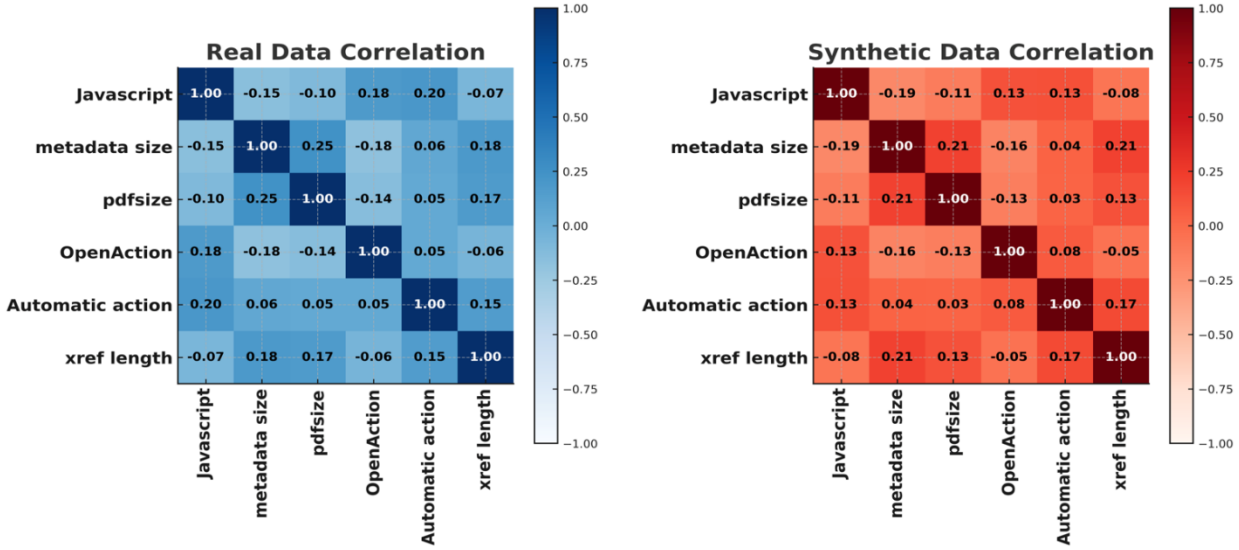

Figure 3: - Pearson Correlation Matrices – Real vs. Synthetic Data

To further explore the global structure, a Principal Component Analysis (PCA) was applied to the concatenated dataset (real + synthetic). As illustrated in Figure 4, the projection onto the first two principal components reveals a substantial overlap in geometric positioning**,** with no emergent clusters or separability between the two data types. This absence of structural divergence supports the notion that CTGAN preserved the latent topological characteristics of the original feature space.

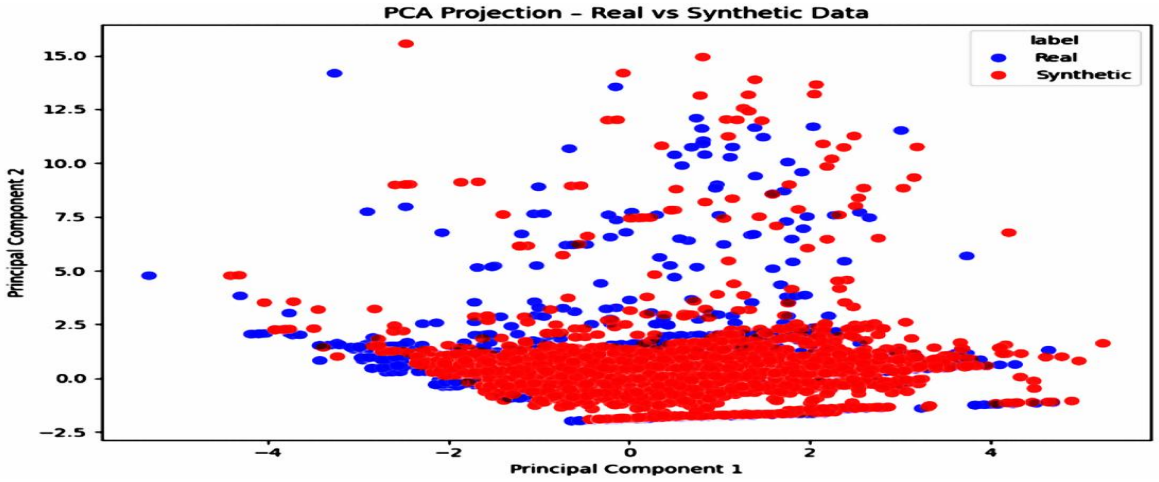

Figure 4: - 2D PCA Projection of Combined Dataset (Real and Synthetic)

Together, these findings attest to the semantic and statistical coherence of the synthetic data. By maintaining both marginal distributions and multivariate associations, the CTGAN-powered augmentation process enhances diversity

in the training set without compromising the interpretability or reliability of the downstream learning algorithms. This enables the overall system to remain performant, even in the presence of initially unbalanced datasets.

Comparative Evaluation of Classification Models Including a Baseline Reference:
The classification module plays a pivotal role in identifying potentially malicious PDF files. It was designed to assess multiple supervised learning algorithms using both the original and CTGAN-augmented datasets, with the aim of selecting the most accurate and generalizable model. This section presents a comparative evaluation of the tested classifiers, ultimately guiding the choice of the most suitable architecture for the subsequent interpretability and diagnostic phases.

Comparative Evaluation of Classifiers on Original and Augmented Datasets:
To assess the robustness and consistency of predictive models, six supervised machine learning algorithms were evaluated on both the original dataset and its CTGAN-augmented counterpart. These included Decision Tree, Random Forest, XGBoost, Support Vector Machine (SVM), Naive Bayes, and a simple Neural Network. In addition, a DummyClassifier was introduced as a baseline reference. This model does not learn from the data but provides a useful benchmark for interpreting the minimum performance threshold that any meaningful classifier should exceed.

Model evaluation was performed using stratified 10-fold cross-validation, a widely accepted technique for obtaining reliable generalization estimates, especially when dealing with class imbalance. Performance was measured using six key metrics: accuracy, precision, recall, F1-score, area under the ROC curve (ROC-AUC), and Matthews Correlation Coefficient (MCC). The results, summarized in Tables 5 and 6, provide a comprehensive overview of model behavior across both datasets.

Table 5: -Cross-validated classification performance on the original dataset (%)

Model	Accuracy	F1-score	Precision	Recall	ROC-AUC	MCC
DummyClassifier	68,14	0,00	0,00	0,00	50,00	0,00
Decision Tree	98.61	97.83	98.49	97.93	98.36	96.81
Random Forest	99.87	99.8	100.0	99.61	99.8	99.71
XGBoost	99.84	99.75	100.0	99.61	99.75	99.64
SVM Classifier	99.37	99.01	99.6	99.61	99.34	98.55
Naïve Bayes	95.47	92.73	96.21	91.11	94.09	89.53
Neural Network	98.61	97.83	98.49	97.93	98.36	96.81

Table 6: -Cross-validated classification performance on the CTGAN-augmented dataset (%)

Model	Accuracy	F1-score	Precision	Recall	ROC-AUC	MCC
DummyClassifier	57,26	0,00	0,00	0,00	50,00	0,00
Decision Tree	97,03	96,43	99,59	95,08	96,59	94,03
Random Forest	99,83	99,81	100	100	99,81	99,66
XGBoost	99,87	99,85	100	100	99,88	99,73
SVM Classifier	98,35	98,04	100	97,85	98,12	96,67
Naive Bayes	86,73	82,35	97,07	73,42	84,92	73,88
Neural Network	97,03	96,43	99,59	95,08	96,59	94,03

As expected, the DummyClassifier returned extremely low scores across all metrics, confirming its role as a non-informative comparator. In contrast, the machine learning models, particularly ensemble methods, demonstrated strong and consistent performance. On the original dataset, Random Forest achieved slightly superior results. However, on the augmented dataset, XGBoost reaches F1 = 99.85%, ROC-AUC = 99.88%, and MCC = 99.73. It outperforms the best non-boosted baseline (SVM, F1 = 98.04%) by +1.81 percentage points in F1 and +1.61 in ROC-AUC. Versus training without augmentation, recall rises from 99.61% to 100% (+0.39 percentage points) and MCC from 99.64 to 99.73 (+0.09), indicating that CTGAN increases minority-class sensitivity while preserving specificity.

These findings underscore the value of CTGAN-based data augmentation in improving class balance and enhancing model learning capacity. The synthetic dataset not only strengthened generalization but also contributed to improved sensitivity, as evidenced by perfect recall scores in XGBoost.Given its exceptional and consistent results, XGBoost trained on the augmented dataset is selected as the reference classifier for subsequent analyses, including interpretability (Section 6.2) and diagnostic visualization (Section 6.3).

Interpretability of the Selected Model: Feature Importance and SHAP Analysis:
To ensure the transparency and explainability of the classification process, we conducted an analysis of feature importance based on the XGBoost model's internal gain metrics. As illustrated in Figure 5, the most influential variable is JavaScript presence, which aligns with known threat signatures in malicious PDFs. Other key predictors include metadata volume, OpenAction triggers, and xref length, highlighting a combination of behavioral and structural indicators.

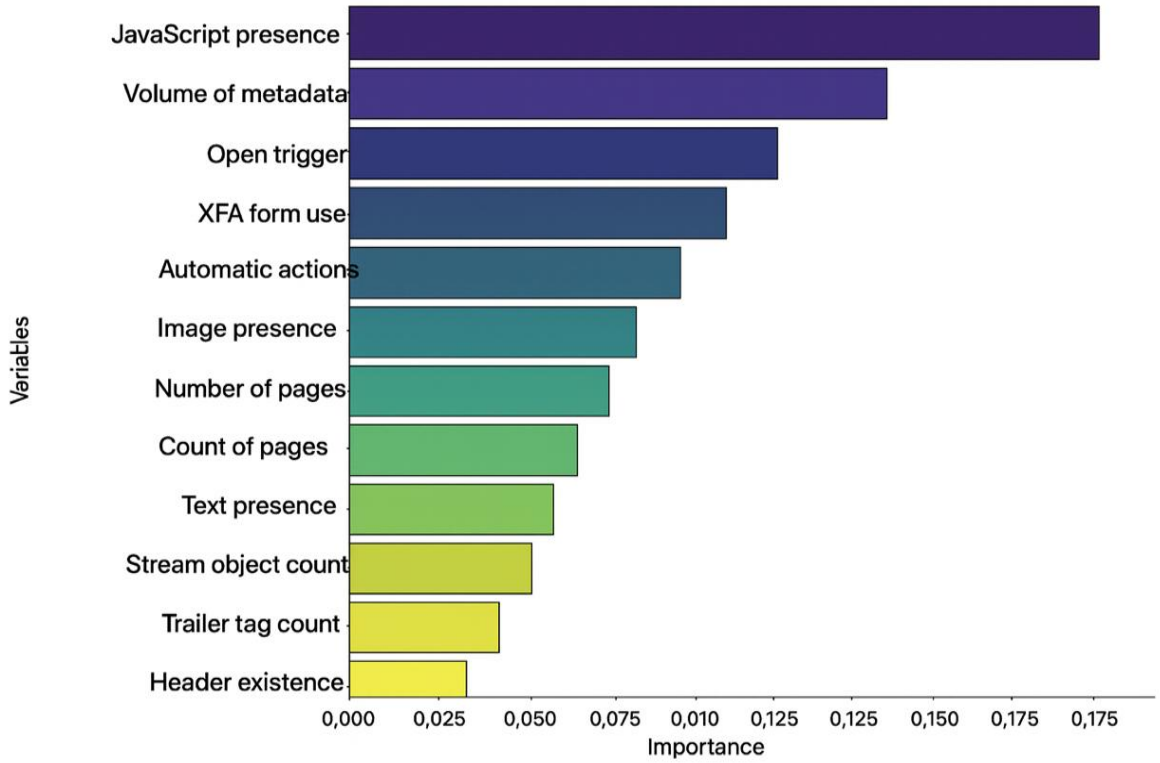

Figure 5: - Feature Importance for the XGBoost Model

In parallel (Figure 6), a SHAP (SHapley Additive exPlanations) analysis was carried out to provide a local and global interpretation of the model's predictions. The SHAP summary plot reveals that high values of JavaScript presence, volume of metadata, and total size of PDF consistently push predictions toward the malicious class, while low values mitigate that likelihood. Conversely, variables such as header existence and text presence exhibit a more nuanced impact, suggesting interactioneffects.

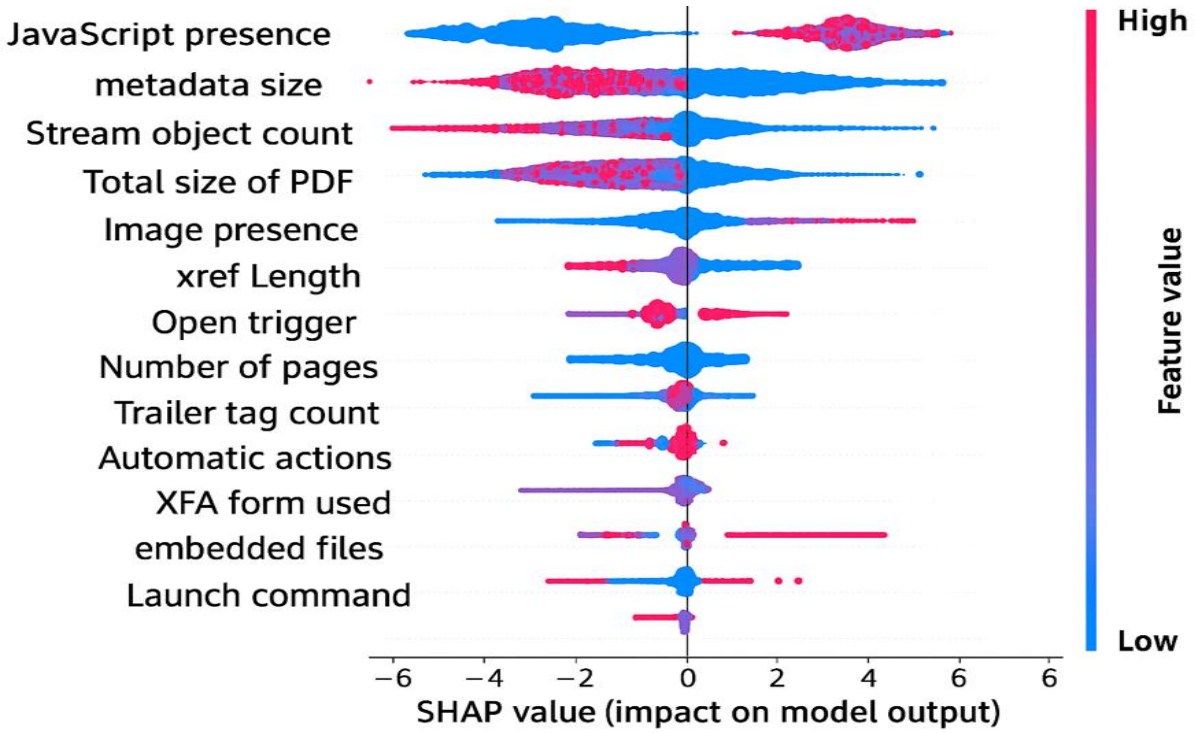

Figure 6: - SHAP Analysis for the XGBoost Model

Collectively, the interpretability analysis confirms that the model's decision-making aligns with domain-relevant indicators. It also enhances user trust by revealing the logical structure behind individual predictions.

Visual Diagnostics and Error Analysis:

The performance of the selected XGBoost model is further validated through visual inspection of its classification behavior. The Receiver Operating Characteristic (ROC) curve**,** shown inFigure 7**,** demonstrates near-optimal separation**,** with an area under the curve of0.9997**,** a hallmark of a highly discriminative classifier.

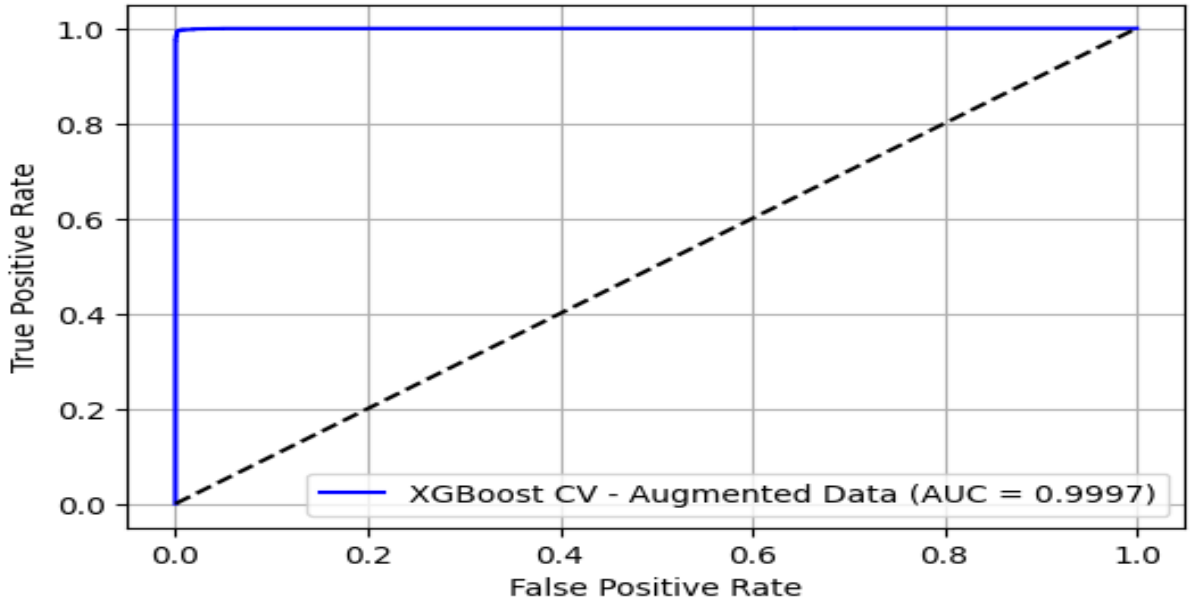

Figure 7: - ROC Curve for the XGBoost Classifier on Augmented Data

Complementing this, the confusion matrixpresentedinFigure 8reports only 15 false positives and 22 false negatives out of 8,752 total instances. This remarkably low error rate reinforces the classifier's precision, sensitivity, and overall reliability.

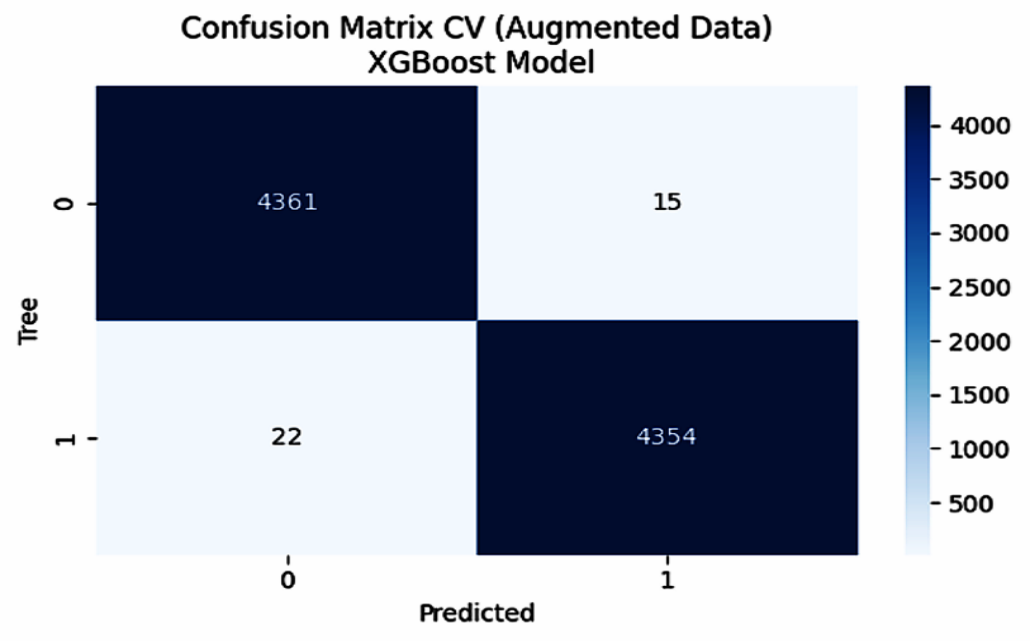

Figure 8: - Confusion Matrix for the XGBoost Classifier on Augmented Data

Taken together, these diagnostic tools provide compelling evidence that the model not only performs well in terms of global metrics, but also maintains operational reliability in fine-grained scenarios. The augmentation strategy via CTGAN, combined with ensemble learning, yields a classification pipeline that is both intelligentandinterpretable, meeting the key demands of modern cybersecurity systems.

Table 7: - Class-wise Metrics on Augmented Data with XGBoost

Class	Precision (%)	Recall (%)	F1(%)	Support
Benign	99.50%	99.66%	99.58%	4,376
Malicious	99.66%	99.50%	99.58%	4,376

As shown in Table 7, performance is very high and nearly symmetric across classes: for Benign, precision 99.50% and recall 99.66%; for Malicious, precision 99.66% and recall 99.50%, giving F1 = 99.58% for both. With 4,376 instances per class, this corresponds to 15 false positives and 22 false negatives. Overall, the XGBoost detector on the augmented dataset shows very low miss and false-alarm rates with no detectable class bias.

Discussion: -

This section provides a critical appraisal of the proposed system, examining its strengths, interpretability, comparative advantages, deployment feasibility, and potential improvements.

Strengths and Interpretability:

One of the key strengths of the proposed system lies in its ability to combine high predictive performance with strong interpretability. The integration of CTGAN for data augmentation, supervised learning, and SHAP-based explanation techniques ensures not only accurate detection but also a transparent decision-making process. This transparency is particularly important in cybersecurity, where threat response must often be explained and justified to stakeholders.

To further contextualize model performance, a DummyClassifier was used as a baseline. Its inability to learn from data resulted in near-zero precision, recall, and F1-score, with ROC-AUC scores at the chance level (50%). This

contrast highlighted the substantial learning capabilities of all other machine learning models, reinforcing the relevance and effectiveness of the proposed pipeline.

Importantly, the SHAP framework consistently revealed key discriminative features such as JavaScript count, OpenAction triggers, and metadata size. These features align with known patterns of malicious PDF behavior [39], [44], [45], further validating the model's trustworthiness. Compared to traditional feature importance scores that offer only global insights, SHAP adds the benefit of local interpretability, enabling analysts to explain individual predictions, a critical asset for integration into human-in-the-loop decision environments [4], [5].

Contribution of Synthetic Data Generation:
Another notable innovation of the framework is the use of CTGAN to overcome class imbalance, a common challenge in malware datasets where benign instances tend to dominate. Traditional resampling techniques such asSMOTE often fail to preserve the complex dependencies that exist in high-dimensional feature spaces. In contrast, CTGAN generates synthetic data that respects the underlying statistical structure of real samples [21].

This augmentation step significantly enhanced the performance of all tested models, with XGBoost benefiting the most. The classifier achieved improved generalization across validation sets, demonstrating that synthetic samples contributed positively to training stability and robustness [2], [46].

Comparison with Existing Methods:
To assess comparative performance, the proposed system was benchmarked against widely used classifiers, including Decision Tree, Random Forest, SVM, Naive Bayes, and a shallow Neural Network. Additionally, a DummyClassifier was included to establish a non-informative reference point. As expected, its results were extremely poor across all metrics, confirming that any meaningful classification must significantly outperform this baseline.

In contrast, XGBoost consistently outperformed all other models, achieving an F1-score of 99.85%, perfect recall and precision, and an AUC-ROC of 99.88% on the augmented dataset. These findings are consistent with prior research highlighting the effectiveness of gradient boosting algorithms in structured data environments for cybersecurity tasks [22], [23].

Unlike deep learning architectures that often sacrifice transparency, the proposed framework offers high interpretability without compromising accuracy. This balance is essential for operational settings, where traceability, explainability, and auditability are often prerequisites for deploying AI solutions [5], [36].

Deployment Feasibility and Operational Relevance:
The system was also designed with practical deployment in mind. Its modular architecture, use of open-source libraries, and compatibility with widely used platforms such as SIEMs enhance its adaptability across various organizational contexts. This makes it suitable not only for research environments but also for integration into production-level cybersecurity infrastructures [3], [20].

Moreover, the system's explainability is a major asset for analysts working in Security Operations Centers (SOCs). As prior studies have shown, transparency in AI-based tools improves trust and promotes adoption by human operators [5], [36]. The relatively low computational cost of the proposed pipeline also supports deployment in environments with limited infrastructure capabilities.

In production we strictly separate augmentation from inference: CTGAN runs offline to rebalance the training set and is never used during detection. Online we only parse the PDF, extract features, and score a lightweight XGBoost model, which is fast enough for near-instant screening on standard CPUs. To scale, we use periodic offline CTGAN refresh with versioning, containerized workers behind a queue with horizontal replication, and drift/health monitoring; SHAP is computed asynchronously or served from cache on demand. We do not claim hard real-time guarantees—latency is driven mainly by parsing, not the model.

Limitations and Future Perspectives:

Despite these strengths, some limitations remain. The current framework operates in offline batch mode, which restricts its use in real-time threat detection. Transitioning to an online learning setup capable of handling streaming data and adapting continuously to evolving attack patterns will be an important direction for future work [47], [48]. Another challenge involves the computational overhead introduced by SHAP explanations. While these interpretations are highly informative, they may become burdensome at scale. Efficient approximation strategies, such as TreeExplainer or surrogate modeling, could help mitigate this issue without compromising interpretability [4], [36].Exploring reinforcement learning and adversarial training mechanisms could further strengthen the system's resilience to novel and adversarial threats [49], [50]. Lastly, although the dataset used in this study reflects real-world conditions, future evaluations on larger and more heterogeneous datasets will be necessary to assess generalizability across different threat landscapes [21], [51].CTGAN training and sample generation carry non-trivial compute and memory costs. We therefore confine augmentation to scheduled offline jobs, triggered at regular intervals or when drift is detected, while only the frozen classifier is served online. Future work will refine latency budgets, explore streaming ingestion for continuous updates, and assess GPU-assisted parsing where it provides a clear benefit.

Conclusion: -

This study presents a robust and modular approach for detecting malicious PDF files, leveraging the complementary strengths of generative data augmentation and supervised machine learning. By strategically integrating these components, the proposed framework addresses three key challenges in cybersecurity: data imbalance, model interpretability, and classification performance.

The use of CTGAN for synthetic data generation proved highly effective in correcting the class imbalance that typically characterizes cybersecurity datasets. This augmentation strategy enriched the training corpus with realistic samples, thereby improving the generalization ability of classifiers in skewed contexts.

Among the six models evaluated, XGBoost consistently demonstrated superior performance, achieving an accuracy of 99.87%, an F1-score of 99.85%, and an MCC of 99.73%. These figures confirm its robustness and adaptability in detecting subtle threat signatures embedded within PDF structures.

To support explainability and user trust, the framework integrates SHAP-based interpretation tools, which reveal how features such as JavaScript presence, OpenAction triggers, and metadata volume influence model predictions. These insights not only enhance the transparency of the system but also provide analysts with actionable indicators grounded in domain knowledge.

In conclusion, the proposed solution combines predictive accuracy, analytical clarity, and operational feasibility, making it a compelling candidate for deployment in diverse cybersecurity environments, including government, academic, and industrial settings.

Future work will explore real-time extensions, active learning for continuous model refinement, and broader validation across heterogeneous datasets. Integrating this system with automated response platforms and threat intelligence services could further enhance its practical relevance and impact in live defense infrastructures.

References: -

[1] L. V. Brown, "Computer Security: Principles and Practice," Pearson Prentice Hall, 2008.
[2] H. He and E. A. Garcia, "Learning from Imbalanced Data," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263–1284, 2009. https://doi.org/10.1109/TKDE.2008.239
[3] J. Z. Kolter and M. A. Maloof, "Learning to Detect and Classify Malicious Executables in the Wild," Journal of Machine Learning Research, vol. 7, no. 1, pp. 2721–2744, 2006. https://doi.org/10.1145/1014052.1014105
[4] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in Advances in Neural Information Processing Systems 30 (NeurIPS), 2017. https://doi.org/10.48550/arXiv.1705.07874
[5] A. B. Arrieta et al., "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges," Information Fusion, vol. 58, pp. 82–115, 2020. https://doi.org/10.1016/j.inffus.2019.12.012
[6] Q. Abu Al-Haija, A. Odeh, and H. Qattous, "PDF Malware Detection Based on Optimizable Decision Trees," Electronics, vol. 11, no. 19, p. 3142, 2022. https://doi.org/10.3390/electronics11193142

[7] A. J. G. De Azambuja et al., "Artificial Intelligence Based Cyber Security in the Context of Industry 4.0 — A Survey," Electronics, vol. 12, no. 8, p. 1920, 2023. https://doi.org/10.3390/electronics12081920
[8] B. Hariharan, R. Siva, S. Sadagopan, V. Mishra, and Y. Raghav, "Malware Detection Using XGBoost Based Machine Learning Models-Review," in 2023 2nd International Conference on Edge Computing and Applications (ICECAA), pp. 964–970, 2023. https://doi.org/10.1109/ICECAA58104.2023.10212327
[9] J. Torres and S. D. L. Santos, "Malicious PDF Documents Detection Using Machine Learning Techniques," in Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018), pp. 337–344, 2018. https://doi.org/10.5220/0006609503370344
[10] B. A. Alabsi, M. Anbar, and S. D. A. Rihan, "Conditional Tabular Generative Adversarial Based Intrusion Detection System for Detecting DDoS and DoS Attacks on the Internet of Things Networks," Sensors, vol. 23, no. 12, p. 5644, 2023. https://doi.org/10.3390/s23125644
[11] M. Buda, A. Maki, and M. A. Mazurowski, "A Systematic Study of the Class Imbalance Problem in Convolutional Neural Networks," Neural Networks, vol. 106, pp. 249–259, 2018. https://doi.org/10.1016/j.neunet.2018.07.011
[12] W. S. Admass, Y. Y. Munaye, and A. A. Diro, "Cyber security: State of the art, challenges and future directions," Cyber Security and Applications, vol. 2, p. 100031, 2024. https://doi.org/10.1016/j.csa.2023.100031
[13] P. Laskov and N. Šrndić, "Static Detection of Malicious JavaScript-Bearing PDF Documents," in Proceedings of the 27th Annual Computer Security Applications Conference (ACSAC), pp. 373–382, 2011. https://doi.org/10.1145/2076732.2076785
[14] D. Maiorca and G. Giacinto, "Clustering-Based PDF Malware Detection Through Dynamic Analysis," Computer Fraud & Security, no. 5, pp. 8–16, 2015. https://doi.org/10.1016/S1361-3723(15)30039-6
[15] U. Bayer, A. Moser, C. Kruegel, and E. Kirda, "Dynamic Analysis of Malicious Code," Journal in Computer Virology, vol. 2, no. 1, pp. 67–77, 2006. https://doi.org/10.1007/s11416-006-0012-2
[16] U. Khadim, M. M. Iqbal, and M. A. Azam, "A Secure Digital Text Watermarking Algorithm for Portable Document Format (PDF)," Mehran University Research Journal of Engineering & Technology, vol. 41, no. 1, pp. 100–110, 2022. https://doi.org/10.3316/informit.263334115228458
[17] U. Premarathne et al., "Hybrid Cryptographic Access Control for Cloud-Based EHR Systems," IEEE Cloud Computing, vol. 3, no. 4, pp. 58–64, 2016. https://doi.org/10.1109/MCC.2016.76
[18] A. Damodaran et al., "A Comparison of Static, Dynamic, and Hybrid Analysis for Malware Detection," Journal of Computer Virology and Hacking Techniques, vol. 13, pp. 1–12, 2017. https://doi.org/10.1007/s11416-015-0261-z
[19] Y. Wang, W. D. Cai, and P. C. Wei, "A Deep Learning Approach for Detecting Malicious JavaScript Code," Security and Communication Networks, vol. 9, no. 11, pp. 1520–1534, 2016. https://doi.org/10.1002/sec.1441
[20] Z. Tzermias, G. Sykiotakis, M. Polychronakis, and E. P. Markatos, "Combining Static and Dynamic Analysis for the Detection of Malicious Documents," in Proceedings of the Fourth European Workshop on System Security, 2011.https://doi.org/10.1145/1972551.1972555
[21] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling Tabular Data Using Conditional GAN," in Advances in Neural Information Processing Systems 32 (NeurIPS), 2019.https://doi.org/10.48550/arXiv.1907.00503
[22] I. Goodfellow et al., "Generative Adversarial Networks," Communications of the ACM, vol. 63, no. 11, pp. 139–144, 2014. https://doi.org/10.1145/3422622
[23] A. S. Dina, A. B. Siddique, and D. Manivannan, "Effect of Balancing Data Using Synthetic Data on the Performance of Machine Learning Classifiers for Intrusion Detection in Computer Networks," IEEE Access, vol. 10, pp. 96731–96747, 2022. https://doi.org/10.1109/ACCESS.2022.3205337
[24] A. Rajkomar, J. Dean, and I. Kohane, "Machine Learning in Medicine," The New England Journal of Medicine, vol. 380, no. 14, pp. 1347–1358, 2019. https://doi.org/10.1056/NEJMra1814259
[25] S. Mendis, P. Puska, and B. Norrving, "Global Atlas on Cardiovascular Disease Prevention and Control," World Health Organization, 2023.
[26] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001. https://doi.org/10.1023/A:1010933404324
[27] S. Zhou, Z. Lu, Y. Liu, et al., "Interpretable Machine Learning Model for Early Prediction of 28-Day Mortality in ICU Patients with Sepsis-Induced Coagulopathy: Development and Validation," European Journal of Medical Research, vol. 29, no. 14, 2024. https://doi.org/10.1186/s40001-023-01593-7
[28] L. Breiman, "Bagging Predictors," Machine Learning, vol. 24, no. 2, pp. 123–140, 1996. https://doi.org/10.1007/BF00058655
[29] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016.

https://doi.org/10.1145/2939672.2939785
[30] L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone, "Classification and Regression Trees," Chapman & Hall/CRC, 2017. https://doi.org/10.1201/9781315139470
[31] C. Cortes and V. Vapnik, "Support-Vector Networks," Machine Learning, vol. 20, no. 3, pp. 273–297, 1995. https://doi.org/10.1007/BF00994018
[32] I. Wickramasinghe and H. Kalutarage, "Naive Bayes: Applications, Variations and Vulnerabilities — A Review with Code Snippets," Soft Computing, vol. 25, no. 3, pp. 2277–2293, 2021. https://doi.org/10.1007/s00500-020-05297-6
[33] B. Ramadhan, Y. Purwanto, and M. Ruriawan, "Forensic Malware Identification Using Naive Bayes Method," in 2020 International Conference on Information Technology Systems and Innovation (ICITSI), pp. 1–7, 2020. https://doi.org/10.1109/ICITSI50517.2020.9264959
[34] C. M. Bishop, "Neural Networks for Pattern Recognition," Oxford University Press, 1995.
[35] S. Haykin, "Neural Networks: A Comprehensive Foundation," 2nd ed., Prentice Hall, 1999.
[36] E. Tjoa and C. Guan, "A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 11, pp. 4793–4813, 2020. https://doi.org/10.1109/TNNLS.2020.3027314
[37] F. Pedregosa et al., "Scikit-Learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[38] C. M. Bishop, "Pattern Recognition and Machine Learning," Springer, 2006.
[39] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," Journal of Machine Learning Research, vol. 13, pp. 281–305, 2012.
[40] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization," Journal of Machine Learning Research, vol. 18, no. 185, pp. 1–52, 2018.https://doi.org/10.48550/arXiv.1603.06560
[41] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," in Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI), 1995.
[42] G. James, D. Witten, T. Hastie, and R. Tibshirani, "An Introduction to Statistical Learning: With Applications in R," Springer, 2013. https://doi.org/10.1007/978-1-4614-7138-7
[43] C. Smutz and A. Stavrou, "Malicious PDF Detection Using Metadata and Structural Features," in Proceedings of the 28th Annual Computer Security Applications Conference (ACSAC), pp. 239–248, 2012. https://doi.org/10.1145/2420950.2420987
[44] S. Dabral, A. Agarwal, M. Mahajan, and S. Kumar, "Malicious PDF Files Detection Using Structural and JavaScript Based Features," in Information, Communication and Computing Technology (ICICCT 2017), Communications in Computer and Information Science, vol. 750, Springer, 2017. https://doi.org/10.1007/978-981-10-6544-6_14
[45] R. Tahsinur, A. Nusaiba, M. Shama, M. H. Fasbeer, and I. H. Muhammad, "Interpreting Machine and Deep Learning Models for PDF Malware Detection Using XAI and SHAP Framework," in 2023 2nd International Conference for Innovation in Technology (INOCON), pp. 1–9, 2023. https://doi.org/10.1109/INOCON57975.2023.10101116
[46] E. Baghirov, "Building Robust Malware Detection Through Conditional Generative Adversarial Network-Based Data Augmentation," Program Systems: Theory and Applications, vol. 15, no. 4, pp. 97–110, 2024. https://doi.org/10.25209/2079-3316-2024-15-4-97-110
[47] T. Jiang, Y. Liu, X. Wu, M. Xu, and X. Cui, "Application of Deep Reinforcement Learning in Attacking and Protecting Structural Features-Based Malicious PDF Detector," Future Generation Computer Systems, vol. 141, pp. 325–338, 2023. https://doi.org/10.1016/j.future.2022.11.015
[48] P. Sangkatsanee, N. Wattanapongsakorn, and C. Charnsripinyo, "Practical Real-Time Intrusion Detection Using Machine Learning Approaches," Computer Communications, vol. 34, no. 18, pp. 2227–2235, 2011. https://doi.org/10.1016/j.comcom.2011.07.001
[49] N. Srndic and P. Laskov, "Practical Evasion of a Learning-Based Classifier: A Case Study," in 2014 IEEE Symposium on Security and Privacy, pp. 197–211, 2014. https://doi.org/10.1109/SP.2014.20
[50] V. Mnih et al., "Human-Level Control Through Deep Reinforcement Learning," Nature, vol. 518, no. 7540, pp. 529–533, 2015. https://doi.org/10.1038/nature14236
[51] J. Gesnouin, S. Pechberti, B. Stanciulescu, and F. Moutarde, "Assessing Cross-Dataset Generalization of Pedestrian Crossing Predictors," in 2022 IEEE Intelligent Vehicles Symposium (IV), pp. 419–426, 2022. https://doi.org/10.1109/IV51971.2022.9827083