PDF features extracted and the description

This table provides a detailed description of the feature extracted by the our PDF parser. The features are grouped into corresponding sub-categories, the description, and significance of each feature is provided for better clarity.

Sl.No.	Features	**Description**	**Significance**
Metadata Fields			
1	author_dot, author_lc, author_len, author_mismatch, author_num, author_oth, author_uc	These fields analyze the Author metadata field in the PDF. They count special characters (dot), lowercase, uppercase letters, numbers, and other characters, along with mismatches compared to other metadata fields.	Discrepancies or anomalies in the Author field indicate attempts to disguise the file's origin or inject malicious metadata.
2	company_mismatch	Indicates if the company name in the metadata does not align with expected patterns or other metadata fields.	Mismatched company details can signal spoofing or tampered metadata.
3	createdate_ and moddate_**	createdate and moddate fields analyze the creation and modification timestamps. Includes dots, mismatches, timezones (tz), timestamps (ts), and version ratios.	Discrepancies or irregularities in timestamps indicate tampering.
4	creator_ and producer_**	Analyze the Creator and Producer metadata fields in similar dimensions as the Author field.	Unusual entries indicate use of uncommon or malicious tools.
5	keywords_* and subject_*	Analyze the Keywords and Subject metadata fields for character distributions, length, and mismatches.	Malicious actors populate these fields with keywords to manipulate indexing or confuse detection systems.
6	pdfid0_* and pdfid1_*	Analyze unique PDF identifiers for character patterns, lengths, and mismatches.	Malicious PDF files use inconsistent or invalid identifiers to evade detection.
Structure and Content Features			
7	count_ and pos_ (e.g., count_endobj, count_stream, pos_image_min)**	Count occurrences and track positions of key PDF components (e.g., objects, streams, images).	Anomalies in counts and positions can suggest embedded malicious payloads (e.g., JavaScript, obfuscated streams).
8	len_obj_ and len_stream_**	Measure the average, minimum, and maximum lengths of PDF objects and streams.	Abnormally large or small object/stream sizes may indicate malicious content, such as embedded scripts or files.
9	box_nonother_types, box_other_only, count_box_ and pos_box_**	Analyze the dimensions and types of bounding boxes (e.g., A4, legal, letter).	Malicious PDFs may have unusual or inconsistent box definitions to bypass print and render settings.
10	count_acroform and pos_acroform_*	Count and analyze positions of AcroForm objects used for forms in PDFs.	AcroForms can embed JavaScript or be manipulated for malicious purposes.
Embedded Objects and Scripts			
11	count_image_ and image_totalpx*	Count occurrences and measure total pixel dimensions of images in the PDF.	Large or numerous images may be used for obfuscation, phishing, or payload embedding.
12	*count_javascript, count_js, and related _obs fields	Count occurrences of JavaScript and observe discrepancies.	Embedded JavaScript is a common method for executing exploits or phishing attacks.
13	count_objstm and count_objstm_obs	Analyze object streams for frequency and consistency.	Object streams can obfuscate malicious content.
Position and Ratio-Based Features			
14	pos_eof_ and count_eof*	Analyze positions and count of the EOF marker in the PDF.	Irregularities in EOF markers may indicate tampered or malformed files.

15	*ratio_ fields**	Calculate ratios like image pixel size to object size or size of streams/pages.	Deviations from expected ratios can hint at hidden content or oversized malicious payloads.
16	*delta_ts and delta_tz*	Measure time differences in timestamps and time zones.	Irregular timing patterns might reveal metadata inconsistencies.
Overall File Characteristics			
17	*version and size*	File version and overall file size.	Malicious PDFs often deviate from standard sizes and may use older versions for compatibility with exploits.
Graph-Based Features: features describe the structural and connectivity properties of the PDF file's internal object structure when modeled as a graph. Nodes represent objects, and edges represent relationships or references.			
18	*avg_degree*	The average degree of nodes in the PDF object graph.	Higher or unusual average degrees may indicate excessive referencing, often seen in obfuscated or overly complex PDFs used for malicious purposes.
19	*avg_clustering_coefficient*	The average clustering coefficient of nodes, showing how interconnected nodes are in the graph.	Malicious PDFs might have specific clustering patterns due to the interconnected nature of objects used in payload obfuscation.
20	*avg_shortest_path*	The average shortest path length between all pairs of nodes in the graph.	Shorter path lengths may suggest dense object referencing, a trait of heavily obfuscated PDFs.
21	*degree_assortativity*	Measures whether high-degree nodes tend to connect to other high-degree nodes (or low-degree nodes).	Anomalous assortativity values could indicate abnormal referencing patterns typical of malicious files.
22	*density*	The overall density of the graph, calculated as the ratio of actual edges to possible edges.	Higher density may suggest excessive object interconnection, often used to obfuscate malicious content.
23	*median_children*	The median number of child nodes per node in the object graph.	An unusually high or low median might indicate abnormal object structures or relationships.
24	*num_edges, num_nodes*	Total number of edges (relationships) and nodes (objects) in the graph.	Malicious PDFs often have unusual object counts or excessive connections due to embedded payloads.
25	*num_leaves*	Number of leaf nodes (nodes with no children) in the graph.	Malicious PDFs might have more leaf nodes if objects are not interconnected or are isolated to hide malicious content.
26	*var_children*	Variance in the number of child nodes across all nodes.	High variance might indicate an unusual distribution of object references, often seen in obfuscated files.
Action and JavaScript Features: features detect potentially malicious behaviors tied to interactive or executable elements.			
27	*/JS and /JavaScript*	Indicators for embedded JavaScript in the PDF.	JavaScript is commonly used in malicious PDFs to execute exploits (e.g., launching payloads, stealing information).
28	*/URI*	Detects Uniform Resource Identifiers (links) embedded in the PDF.	Malicious PDFs often contain links leading to phishing websites or malicious downloads.
29	*/Action, /AA, /OpenAction*	Indicators for actions or automatic actions triggered upon opening the PDF.	These features can execute malicious payloads without user interaction.

30	*/launch and /submitForm*	Commands to launch external applications or submit form data.	These commands can be exploited to execute arbitrary code or exfiltrate data.
Form and Multimedia Features: These features focus on embedded forms, multimedia elements, and specific data streams.			
31	*/Acroform and /XFA*	Detects the presence of AcroForm and XML Forms Architecture (XFA).	Forms can be used to embed malicious JavaScript or steal user data.
32	*/JBig2Decode*	Detects the use of the JBIG2Decode filter, typically used for image compression.	Exploits targeting JBIG2Decode vulnerabilities are known, making its presence a potential red flag.
33	*/Colors*	Indicates the use of color-related objects or properties.	Irregularities in color definitions indicate attempts to obfuscate malicious content.
34	*/Richmedia*	Detects embedded rich media, such as videos or interactive content.	Rich media can be exploited to execute malicious payloads.
Structural Features: features identify critical structural components of PDFs.			
35	*/Trailer, /Xref, /Startxref*	Indicators for the trailer dictionary, cross-reference table (Xref), and start of the cross-reference section (Startxref).	Manipulating these structures is a common tactic for hiding malicious content or creating malformed PDFs to exploit parsers.