A Comprehensive Survey on Security Features and Vulnerabilities in Data Science Tools

Fatema Islam Meem *meem7822@vandals.uidaho.edu*
Department of Computer Science
University of Idaho
Moscow, 83844, USA

Imran Hussain Mahdy *mahd3488@vandals.uidaho.edu*
Department of Chemical and Biological Engineering
University of Idaho
Moscow, 83844, USA

Sabiha Jannath Tisha *tish3434@vandals.uidaho.edu*
Department of Computer Science
University of Idaho
Moscow, 83844, USA

Shahidur Rahoman Sohag *soha1641@vandals.uidaho.edu*
Department of Computer Science
University of Idaho
Moscow, 83844, USA

Abstract

Data science tools have grown quickly, changing many industries by allowing advanced data analysis, predictive models, and more intelligent decisions. However, their rapid development has also introduced significant security challenges and vulnerabilities. This study investigates the security features and weaknesses commonly found in widely used data science tools. The analysis focuses on key security mechanisms and identifies frequent vulnerabilities. The research aims to comprehensively comprehend the security landscape within the data science domain by examining these aspects. The findings underline the critical need for robust security protocols to safeguard data integrity, confidentiality, and privacy in data-driven processes. This work aims to guide users in adopting better security strategies and enhancing the overall safety of their data science workflows.

Keywords: Data Science Security, Data Science Tools, Data Protection, Secure Data Analysis.

1. INTRODUCTION

The rapid growth of data in recent years has made data science a crucial field, helping organizations gain insights and make better decisions. Tools and platforms like Python, R, Jupyter, and various machine learning libraries are now essential for handling and analyzing large datasets (Nguyen et al., 2019). These tools enable advanced data processing, predictive modeling, and visualization, transforming industries such as healthcare, finance, marketing, and many others (Sohagetal., 2024). However, the growing dependence on these tools presents significant security concerns. Since data science tools often handle sensitive and critical information, ensuring their security is vital (Tian2020). Security breaches, unauthorized access, and other threats can result in financial losses, legal issues, and reputational damage. Protecting not only the data but also the trust and credibility of an organization is essential (Pereira, Silva, and Orvalho 2020) (Reece et al., 2023). Data science tools differ in their architecture, functionality, and security needs. While some incorporate strong security measures, others may have vulnerabilities that malicious actors can exploit (Alharbi 2020). Security features include data encryption, access controls, audit logging, and secure coding practices.These safeguards aim to maintain data integrity, confidentiality, and availability, allowing data science operations to run securely (Almalki and Song2020) (Shukla etal.,2022). Moreover, while promoting innovation, the open-source nature of many data science tools can make it easier for vulnerabilities to spread if not appropriately managed (I. Martinez, Viles, and Olaizola 2021).

This survey explores the security features and vulnerabilities of widely used data science tools. It reviews existing literature and examines real-world case studies to identify the best practices for secure usage. The survey analyzed peer-reviewed papers, case studies, and reports to ensure a comprehensive review. Data was collected from reputable sources, focusing on publications from 2016 to 2024. This approach provides the relevance of findings to current security challenges.

Research Questions:

1. What are the most common vulnerabilities in widely used data science tools?
2. How effective are the existing security mechanisms in addressing these vulnerabilities?
3. What strategies and best practices can be employed to balance usability and security in data science workflows?

The study serves as a resource for data scientists, developers, and security experts aiming to enhance the security of their tools and workflows. Beyond identifying features and weaknesses, the study analyzes potential enhancements to prevent attacks. It also explores the balance between usability and security, which is another critical gap. Overly complex security mechanisms can delay user adoption, while more straight forward configurations may leave systems vulnerable. Addressing these issues requires regular updates, robust access controls, and improved user education on secure configurations. By addressing these challenges, this study contributes to developing safer data science practices and tools, offering better protection for valuable data.

2. OVERVIEW OF DATA SCIENCE TOOLS

Data science has transformed the way organizations handle and analyze data, allowing them to make more informed decisions. Numerous tools have been developed to support this transformation, each focusing on specific stages of the data science process (Haq et al., 2020) (Baviskar et al., 2021). These tools include programming languages, specialized software packages, and platforms that facilitate data collection, processing, analysis, and visualization. Data scientists can efficiently manage complex workflows and extract meaningful insights from large datasets using these tools. To better understand the functionalities and applications of these tools, we have categorized them into three main groups. This categorization helps highlight their roles in various stages of the data science lifecycle, offering a clearer perspective on how they contribute to solving diverse analytical challenges.

2.1 Proprietary Tools

Proprietary data science tools are commercial software developed and owned by companies (Gorelik 2019). These tools are generally licensed to users for a fee (Eisenmann 2008). They often come with user-friendly interfaces, extensive documentation, dedicated customer support, and regular updates. While these tools are highly efficient and reliable, they can be costly for individual users or small teams. Additionally, they may limit customization and integration with other tools.

2.1.1 Statistical Analysis System (SAS)

SAS is a comprehensive software suite for advanced analytics, business intelligence, data management, and predictive analytics (Pope 2017). It provides tools for data manipulation, statistical analysis, and reporting. SAS offers various statistical procedures, such as linear regression, variance analysis, and high-performance model selection for significant data sources

(Svolba 2017). Known for its rigorous testing and quality assurance, it delivers reliable results. SAS is widely used in healthcare, finance, and government industries that require strict data management and compliance standards. Its cross-platform support and scalability enable it to handle diverse computing environments and data sources (Marasinghe and Koehler 2018).

2.1.2 MATLAB

MATLAB is a high-performance programming language and interactive environment designed for numerical computation, data analysis, and visualization. It allows users to solve problems using familiar mathematical notation and offers extensive toolboxes for specialized fields like signal processing, control systems, and machine learning (Martinez, Martinez, and Solka 2017). MATLAB is particularly popular in engineering and scientific domains due to its powerful matrix manipulation capabilities. It also integrates well with other programming languages like Python and C/C++, making it adaptable for larger projects (Ciaburro 2017) (Yang et al., 2022).

2.1.3 Tableau

Tableau is a leading data visualization tool that enables users to create interactive and shareable dashboards (Patel 2021). It connects to various data sources, such as databases, spreadsheets, and cloud services, and allows users to visualize data using charts, graphs, and maps (Mittal and Raheja 2024). Tableau’s drag and drop interface makes it accessible to users without technical expertise. It is widely used in business intelligence for generating insights and supporting data-driven decisions (Islam et al., 2020).

2.1.4 Alteryx

Alteryx is a data analytics platform that simplifies data preparation, blending, and analysis through a drag-and-drop interface. Users can create workflows to integrate, transform, and analyze data without extensive coding knowledge (Buratti et al., 2023). Alteryx supports a variety of operations, including data wrangling, predictive analytics, geospatial analysis, and reporting (Bilokon, Bilokon, and Amen 2023). It allows the inclusion of R and Python code in workflows and offers pre-built predictive models, making it a versatile choice for industries like finance, healthcare, and retail (Serra, Estima, and Silva 2023).

2.1.5 Qlik

Qlik is a business intelligence platform that provides data visualization, exploration, and reporting tools. It offers products like Qlik Sense and Qlik View, which enable users to create interactive dashboards and reports (Ghaffar 2020). Qlik’s associative data model connects multiple data sources, allowing users to explore relationships between data points (Labbe et al., 2019). It supports self-service analytics, enabling users without technical expertise to create visualizations. Qlik is known for handling large datasets effectively, making it a popular choice for analytics and reporting (Purgindla 2018).

2.1.6 TIBCO Spotfire

TIBCO Spotfire is a platform for data visualization and analytics. It supports real-time data analysis and predictive analytics, enabling users to create interactive dashboards and explore trends (Xavier 2013). Spotfire integrates with various data science tools and libraries, offering capabilities for machine learning and statistical modeling (Roth-Dietrich, Groschel, and Reiner 2023). It is widely used in industries like healthcare, energy, and manufacturing for its ability to provide actionable insights from complex data (Morabito and Morabito 2016).

2.2 Open Source Tools

Open-source data science tools are freely available, allowing users to use, modify, and share their source code. These tools offer transparency, flexibility, and cost-effectiveness, benefiting from active community contributions. However, they often require technical expertise and may lack the user-friendly features and official support found in proprietary tools (Llerena et al., 2019).
Some popular open-source tools:

2.2.1 Python

Python is a versatile and widely used open-source programming language known for its simplicity

(Elhalid, Alhelal, and Hassan 2023). It is highly favored in data science due to its rich ecosystem of libraries (Ranjan et al., 2023). Libraries like NumPy support numerical computations and matrix operations, while Pandas simplifies working with structured data (VanderPlas 2016) (Molin 2021). Scikit-learn provides tools for machine learning, including algorithms for classification, regression, and clustering (Paper 2019). For data visualization, libraries such as Matplotlib and Seaborn enable the creation of detailed and interactive visuals (Sial, Rashdi, and Khan 2021) (Meem and Mishu 2023). Python's readability and extensive community support make it a go-to tool for data scientists (Raschka, Patterson, and Nolet 2020) (Mishu et al., 2021).

2.2.2 R

R is a programming language specifically designed for statistical computing and graphics (Pavlenko et al., 2022). It offers numerous packages for data analysis and visualization. For instance, "ggplot2" is widely used for creating complex visualizations, while "dplyr" and "tidy" simplify data manipulation and cleaning (Wickham, Rundel, and Grolemund 2023). R excels in statistical modeling and machine learning, making it popular in academia and research settings (Ramasubramanian and A. Singh 2017).

2.2.3 Apache Hadoop

Apache Hadoop is a framework for distributed processing of large datasets. It enables scalable data storage and processing across multiple machines (Zahra and Ashif 2020). Hadoop is highly valued in big data applications for its fault tolerance and ability to handle structured, semi-structured, and unstructured data (Bhosale and Gadekar 2014). Its ecosystem includes tools like "Apache Spark," "Hive," and "HBase," which expand its capabilities for data processing and analytics (Hussein 2020) (Sewal and Singh 2021).

2.2.4 Jupyter Notebook

Jupyter Notebook provides an interactive environment for computational tasks. It allows users to create and share documents that combine code, equations, visualizations, and text (Weiss 2020). This flexibility makes it a powerful tool for data scientists working on diverse tasks, including data cleaning, statistical modeling, machine learning, and data visualization. It has over 40 programming languages, such as Python, R, Julia, and Scala, making it highly versatile for various data science workflows (Lavin 2016). Its advanced interface, JupyterLab, offers enhanced flexibility and extensibility, enabling users to handle complex workflows in data science, scientific computing, and machine learning (Gupta 2021). Documents created in Jupyter Notebook can be easily shared through email, Dropbox, GitHub, or the Jupyter Notebook Viewer, promoting seamless collaboration on data science projects (Mendez et al., 2019). This accessibility and collaborative functionality have made Jupyter Notebook a widely used tool in the data science community.

2.2.5 ApacheSpark

Apache Spark is designed for processing large-scale data efficiently. It is built to handle big data workloads by using in-memory caching and optimized query execution, which allows it to perform analytics quickly on datasets of any size (Uta et al., 2022). Spark supports multiple programming languages, including Java, Scala, Python, and R, making it adaptable for various user needs. It offers a unified analytics engine for processing large datasets (Gupta and Kumari 2020) (Omar and Jumaa 2019). Spark includes several key components: "SparkSQL" for running interactive queries, "SparkStreaming" for analyzing real-time data, "MLlib" for machine learning tasks, and "GraphX" for graph processing (Ankam 2016). One of Spark's strengths is its ability to handle batch processing and real-time analytics, making it a flexible tool for diverse industries. It is commonly used in finance, healthcare, manufacturing, and retail sectors, where fast and efficient data processing is essential for decision-making (Aziz, Zaidouni, and Bellafkih 2018).

2.2.6 MySQL

MySQL is a popular open-source relational database management system (RDBMS) widely used for its reliability, scalability, and user-friendly nature (Rawat, Purnama, et al., 2021). It supports storage engines, such as "InnoDB" and "MyISAM." It offers advanced features like "ACID" compliance for transaction reliability, foreign keys for maintaining data integrity, and full-text indexing for efficient searches. MySQL HeatWave is a fully managed database service that integrates transactions, real-time analytics, and machine learning, simplifying workflows by eliminating the need for complex ETL processes (Ogutu 2016). This makes MySQL suitable for various applications, including web development, data warehousing, and business intelligence. Tools like “PowerBI,” “Tableau,” and “Datapine” can also be used alongside MySQL to enhance its analytical capabilities (Ghaffar 2020).

2.2.7 NoSQL Databases (MongoDB, Cassandra)

NoSQL databases are built to manage large volumes of unstructured or semi-structured data effectively (Chang and Chua 2019). They are designed to provide flexibility and scalability for modern applications. Two widely used NoSQL databases are “MongoDB” and “Cassandra.” “MongoDB” is a document-oriented database that stores data in a JSON-like BSON format. This structure makes it highly flexible and scalable, allowing it to handle dynamic data and adapt to changing requirements. It is particularly suitable for applications that demand high performance and horizontal scalability, such as content management systems or real-time analytics (Pokorny 2020). “Cassandra” is a distributed database that processes large amounts of data across multiple servers. It eliminates single points of failure, ensuring high availability and fault tolerance. This makes it ideal for real-time big data applications, such as e-commerce, social media, and financial services, where uninterrupted service is critical (Wahid and Kashyap 2019). Both databases are valuable tools for handling the growing demands of data-intensive applications in various industries.

2.2.8 PostgreSQL

PostgreSQL supports various data types, including JSON, XML, and arrays, making it suitable for diverse applications. Advanced features such as full-text search, indexing, and transactional integrity ensure efficient and secure data management (Rosid 2017). One of PostgreSQL’s key strengths is its extensibility. Users can define custom data types, operators, and functions, allowing the database to be tailored to specific needs. This capability makes PostgreSQL highly versatile and adaptable for various industries. PostgreSQL is widely used in scenarios that require complex queries and high data integrity. Examples include financial systems, data warehousing, and geospatial applications. Its robust architecture and advanced features make it a preferred choice for managing critical and complex datasets (Sveen 2019).

2.2.9 Konstanz Information Miner (KNIME)

KNIME allows users to design data workflows through a simple drag-and-drop interface, making it easy to use even for those with limited technical expertise (Acito 2023). The platform supports various data sources and provides data preprocessing, analysis, and visualization tools. This makes it highly effective for preparing and exploring data. Additionally, KNIME integrates with various machine learning libraries and tools, enhancing its capabilities for advanced data science projects (Fillbrunn et al., 2017). KNIME is widely used in academic and industrial environments due to its flexibility and user-friendly design. It is a valuable resource for data scientists working on diverse projects, from fundamental analysis to complex machine learning workflows.

2.2.10 RapidMiner

RapidMiner is a comprehensive data science platform that supports tasks like data preparation, machine learning, deep learning, text mining, and predictive analytics (Kotu and Deshpande 2014). It provides a user-friendly visual workflow designer, enabling users to create and deploy models without writing code. The platform works with various data sources and offers tools for data preprocessing, model validation, and performance evaluation. This makes it a versatile choice for handling different stages of the data science process. RapidMiner is widely used in finance, healthcare, and manufacturing because it manages complex data science workflows efficiently. Its simplicity and flexibility appeal to both experienced data scientists and those new to the field.

2.3 Cloud-Based Tools

Cloud-based data science tools are hosted on remote servers and accessed via the Internet.

These tools utilize cloud infrastructure's scalability, flexibility, and computing power, making them ideal for handling large-scale data science tasks. They provide several advantages, such as on-demand scalability, reduced need for managing physical infrastructure, and access to high-performance computing resources (Hajare et al., 2021). This allows users to focus on data analysis without worrying about hardware limitations. However, cloud-based tools may involve recurring costs based on usage, potential fees for data transfer, and reliance on the services and policies of the cloud provider. These factors should be considered when adopting cloud solutions for data science. Such tools include various cloud platforms and services specifically designed for analytics and machine learning.

2.3.1 Google Cloud Platform (GCP)

Google Cloud Platform (GCP) provides a wide range of cloud computing services that operate on the same infrastructure Google uses internally. These services include tools for data storage, machine learning, and data analysis, all backed by built-in solid security features (Lakshmanan 2022) (Wijaya 2022). GCP supports diverse data science activities through services like "BigQuery", a fully managed data warehouse designed for fast and efficient analytics, and "TensorFlow", a powerful open-source library for machine learning and deep learning. For big data processing, GCP offers "Cloud Dataproc", which enables users to run scalable "Apache Spark" and "Apache Hadoop" clusters on the cloud (Ramuka 2019). Additional services include "Cloud Dataflow" for processing batch and streaming data, "Cloud Dataprep" for cleaning and preparing data, and "Cloud Datalab", which supports interactive data analysis and machine learning tasks. These tools make GCP a versatile platform for handling complex data science workflows efficiently and cost-effectively (Sukhdeve and Sukhdeve 2023b) (Roy, Banerjee, and Bhardwaj 2021).

2.3.2 Google Colab

Google Colaboratory, often called Google Colab, is a cloud-based platform that allows users to write and execute Python code in a Jupyter Notebook environment (Sukhdeve and Sukhdeve 2023a). It is specifically designed for machine learning and data science tasks, offering free access to advanced computational resources like GPUs and TPUs, which enhance processing power (Kimm, Paik and Kimm 2021). Google Colab eliminates the need for complex local setup and configuration, making it easy for users to start coding quickly. It integrates smoothly with Google Drive, allowing users to save, access, and share their work effortlessly (Ambrosio-Cestero, Ruiz-Sarmiento, and Gonzalez-Jimenez 2023). The platform supports popular Python libraries, such as "TensorFlow", "PyTorch", and "Matplotlib", widely used for machine learning, data analysis, and visualization tasks (Antiga, Stevens, and Viehmann 2020). One of its standout features is the ability for multiple users to collaborate on the same notebook in real-time.

2.3.3 BigML

BigML provides various tools for tasks such as classification, regression, clustering, anomaly detection, and time series forecasting. The platform features an easy-to-use web interface, APIs, and command-line tools like BigMLer, which help automate machine learning workflows (Sasikala 2017). It integrates with various applications and services, including Google Sheets, Zapier, and Alexa, making it versatile and adaptable for different use cases (Chandra et al., 2022) (Elshawi et al., 2018). Key features include model interpretability, the ability to export models, and collaboration tools, ensuring accessibility for users with varying levels of technical expertise.

2.3.4 Databricks

Databricks is a collaborative data analytics platform for data engineering, data science, and machine learning tasks. It is built on Apache Spark and provides a shared environment where data scientists and engineers can work together on data projects (Pala 2021). The platform supports a variety of data sources and includes tools for data ingestion, transformation, and analysis. Databricks also offers advanced machine learning features, such as "AutoML" for automating model development and "MLflow" for managing the entire machine learning lifecycle (L'Esteve 2022). Its collaborative notebooks enable users to write and execute code in multiple languages, including Python, R, Scala, and SQL, making it versatile for diverse workflows. It is a popular choice for big data analytics and machine learning applications (Fowdur et al., 2018).

2.3.5 IBM Watson Studio

IBM Watson Studio provides tools for data preparation, machine learning model building, and deployment. The platform integrates with other IBM services, such as “Watson Machine Learning” and “Watson Knowledge Catalog”, to create a unified environment for managing data science workflows (Cecil and Soares 2019). Watson Studio supports multiple programming languages, including Python, R, and Scala, making it versatile for user preferences. It also offers tools for data visualization, model management, and team collaboration, ensuring an efficient workflow for complex projects (Miller 2019). This makes IBM Watson Studio a comprehensive solution for organizations looking to enhance their data science capabilities.

3. SECURITY FEATURES IN DATA SCIENCE TOOLS

The protection of sensitive data is essential in data science. As organizations gradually rely on data-driven decision-making, the tools used for data collection, processing, and analysis must be equipped with robust security features (Mahdy et al., 2024). These security measures are essential for safeguarding data from unauthorized access, breaches, and other potential threats that could compromise its integrity and confidentiality. This section explores the security features of various proprietary, open-source, and cloud-based data science tools, providing a detailed overview of how these tools safeguard data throughout the data lifecycle.

3.1 Encryption

Encryption is widely implemented to secure data at rest (stored data) and in transit (data being transmitted). It ensures that even if the data is intercepted or accessed by unauthorized parties, it remains unreadable without proper decryption keys (Hazra et al., 2024). Many tools incorporate advanced encryption protocols like SSL (Secure Sockets Layer) and TLS (Transport Layer Security) to enhance data protection. For example, Tableau and SAS leverage SSL/TLS encryption during data transmission. These protocols safeguard sensitive information, such as financial records and medical data, ensuring its confidentiality while moving between systems (Cloud Security Report: Tableau Cloud Security in the Cloud 2024). Similarly, Alteryx and Qlik extend this security by implementing encryption for data at rest and in transit. This dual approach prevents unauthorized access to stored data and secures its transmission over networks (Henry, Heath, and Jong 2022) (Empowering Organizations With Solutions They Can Trust n.d.). Open-source tools also prioritize encryption. Jupyter Notebook supports HTTPS, a widely used protocol that encrypts data exchanges between users’ browsers and the notebook server. This prevents attackers from intercepting or tampering with sensitive data during active sessions (Cao 2024). Additionally, Apache Hadoop, a distributed data storage and processing tool, offers encryption at rest and in transit, securing data exchanges within its clusters (Parmar et al., 2017). Cloud-based platforms like Google Cloud Platform (GCP) provide robust encryption features as part of their infrastructure. By default, GCP encrypts data stored on its servers, ensuring that sensitive information remains secure even if storage media are physically accessed. For data in transit, GCP uses TLS, which protects information as it moves between different services or users (Security overview 2024).

3.2 Role-Based Access Control (RBAC)

Role-Based Access Control (RBAC) is a fundamental security feature designed to restrict user access based on organizational roles. By assigning specific permissions to users according to their responsibilities, RBAC ensures that individuals can only access the data and functionalities necessary for their tasks. This approach significantly reduces the risk of accidental or intentional misuse of sensitive information. Proprietary tools like SAS and Tableau utilize RBAC to manage user permissions efficiently (Klein, Tyler, and Fields 2022). Administrators can define and assign roles, ensuring only authorized individuals can access specific data or features. For instance, sensitive datasets in SAS can be restricted to particular roles. At the same time, Tableau ensures that users can interact with dashboards or data sources only if they have the appropriate permissions (Cloud Security Report: Tableau Cloud Security in the Cloud 2024). Similarly, tools such as TIBCO Spotfire and Qlik implement RBAC to control data exposure. In these platforms, administrators can establish fine-grained permissions for users, limiting their access to only the data and analytics tools required for their roles. This prevents unauthorized users from viewing or modifying critical information, thus maintaining data security (Security Advisory regarding TIBCO Spotfire 2023) (Empowering Organizations with Solutions They Can Trust n.d.). Open-source tools also incorporate RBAC principles. Frameworks like Django and Flask in Python provide built-in capabilities to define user roles and enforce access controls. These frameworks enable developers to secure web applications by ensuring users can only access authorized pages, functions, or data based on their roles (Ablahd 2023). Cloud-based platforms such as BigML and Databricks offer advanced RBAC systems. These platforms allow organizations to customize permissions according to specific projects, datasets, or tools. For example, BigML ensures that machine learning models and data are accessible only to designated team members (Security guide - Azure Databricks 2024). At the same time, Databricks restricts access to clusters, notebooks, and datasets based on user roles (Zhai et al., 2016).

3.3 Audit Logging

Audit logging is an essential security feature that tracks user activities, including data access, modifications, and other interactions within a system. By maintaining a comprehensive record of these activities, audit logs enable administrators to monitor user behavior, detect suspicious actions, and respond promptly to potential security incidents (Edwards 2024). This feature is critical in ensuring transparency, accountability, and compliance with security standards. Proprietary tools like SAS and Tableau utilize robust audit logging mechanisms to enhance security. SAS provides detailed audit trails that record user activities, such as accessing or modifying data. These logs help administrators perform forensic analysis in case of a security breach, enabling them to identify and address vulnerabilities effectively. Similarly, Tableau tracks user behavior through audit logging, which allows administrators to monitor activities and take corrective measures when security incidents are detected. For instance, unauthorized access attempts or unusual data usage patterns can be identified and mitigated swiftly (Cloud Security Report: Tableau Cloud Security in the Cloud 2024). Open-source tools also incorporate logging mechanisms to enhance transparency. Apache Spark includes logging features that monitor system and user activities, ensuring accountability in distributed computing environments (Boros, Lehotay-Ke´ry, and Kiss 2023). Jupyter Notebook, widely used for interactive coding, employs logging to track user sessions and actions, which helps prevent unauthorized access and maintains operational transparency (Cao 2024). Cloud-based platforms like IBM Watson Studio extend audit logging to comply with industry security standards. Watson Studio's logging system ensures that all user activities are securely recorded, providing transparent data access and interaction trails. This is particularly valuable for organizations handling sensitive data, as it ensures both regulatory compliance and operational security (Ombiro 2016).

3.4 Other Security Features

Beyond the core security measures of encryption, RBAC, and audit logging, data science tools employ various additional features to enhance data protection and system integrity. These features address a broader range of security challenges, ensuring comprehensive safeguarding of sensitive information.

3.5 Authentication Mechanisms

Authentication mechanisms are pivotal in preventing unauthorized access. Many tools implement advanced authentication systems to verify the identity of users. For instance, SAS and Tableau support Single Sign-On (SSO), allowing users to authenticate once and access all connected resources without re-entering credentials. This streamlines access and reduces vulnerabilities related to password misuse (Cloud Security Report: Tableau Cloud Security in the Cloud 2024). Jupyter Notebook incorporates token-based authentication, requiring a unique token for server access, effectively preventing unauthorized users from gaining entry to active sessions (Cao 2024). Additionally, IBM Watson Studio implements multi-factor authentication (MFA), which requires users to verify their identity using two or more factors, such as a password and a mobile app, providing an added layer of security (Ombiro 2016).

3.6 Secure Configurations

Secure configurations are another vital aspect of data protection. Many tools emphasize secure default settings and allow administrators to fine-tune configurations to align with organizational security requirements. For example, Apache Hadoop supports Access Control Lists (ACLs), allowing administrators to define granular permissions for files and directories, ensuring that data is accessible only to authorized users (Parmar et al., 2017). Similarly, Databricks provides automated security configurations such as secure cluster policies, ensuring that environments are deployed with pre-configured security standards, reducing the likelihood of configuration errors (Security and compliance guide 2024).

3.7 Data Masking

Some tools employ data masking to protect sensitive information by substituting it with fictitious, realistic data. This ensures that the actual sensitive details are not exposed even if unauthorized access occurs. For instance, Alteryx supports data masking to safeguard critical data during analysis and reporting (Henry, Heath, and Jong 2022). Similarly, TIBCO Spotfire implements dynamic masking, which controls the visibility of data for different users based on their roles, aligning with privacy and compliance requirements (Security Advisory regarding TIBCO Spotfire 2023).

3.8 Vulnerability Management

Vulnerability management is another essential feature aimed at mitigating risks posed by software flaws. Tools like Tableau regularly monitor and release updates to address vulnerabilities, such as the Apache Log4j2 vulnerability, ensuring systems remain secure against emerging threats ("Apache Log4j2 vulnerability (Log4shell)" 2022). Apache Spark takes a proactive approach by incorporating runtime monitoring tools to detect and resolve potential security issues during operations, minimizing the risk of escalation (Boros, Lehotay-Ke´ry, and Kiss 2023).

3.9 Compliance with Standards

Compliance with industry standards is integral for tools handling sensitive data, particularly in regulated industries. Platforms like IBM Watson Studio adhere to regulations such as GDPR, HIPAA, and ISO standards, ensuring data security and privacy compliance for its users (Ombiro 2016). Similarly, GCP aligns with frameworks like SOC 2, PCI DSS, and FedRAMP, providing a secure environment for data science workflows and demonstrating its commitment to regulatory compliance (*Security overview 2024*).

4. COMMON VULNERABILITIES IN DATAS CIENCE TOOLS

Data science tools are designed to process and analyze extended amounts of data efficiently. However, their complexity and overall usage make them susceptible to various vulnerabilities. Malicious actors can exploit these weaknesses to compromise data integrity, confidentiality, and system functionality. This section examines common types of vulnerabilities in data science tools and highlights notable case studies to illustrate their impact and mitigation strategies.

4.1 Injection Attacks (SQL and NoSQL)

Injection attacks occur when an attacker exploits improper handling of user inputs in database queries. These attacks are a significant threat to SQL and NoSQL databases, as they allow unauthorized access to sensitive information, data manipulation, or even complete system compromise. SQL injection is an attack where malicious SQL statements are inserted into input fields to exploit vulnerabilities in query construction (Galluccio, Caselli, and Lombari 2020). For example, SAS systems have been found vulnerable to SQL injection due to insufficient input validation, where attackers manipulated input to execute unauthorized database commands (Sadotra and Sharma 2017). If input data is directly incorporated into SQL queries without adequate sanitization, attackers can gain access to sensitive information or alter the database structure. NoSQL injection is a similar risk for NoSQL databases, such as MongoDB and Cassandra. These databases often process queries in JSON or other flexible formats, which attackers can manipulate. For instance, poorly handled query structures in MongoDB allow attackers to bypass authentication mechanisms or access unauthorized data by injecting malicious JSON queries (NoSQL injection 2024). The flexible nature of NoSQL query languages can make it more challenging to enforce strict security controls, increasing the potential for exploitation. Both types of injection attacks highlight the importance of input validation, query parameterization, and robust access controls.

4.2 Misconfigurations

Misconfigurations in tools or infrastructure components are a common cause of security vulnerabilities. These errors can provide attackers with entry points to exploit systems, leading to unauthorized access, data breaches, or code execution. In Apache Hadoop, misconfigured access controls in the YARN Resource Manager services have posed serious security risks (Tall and Zou 2023). In some cases, attackers have exploited these misconfigurations to submit unauthorized applications to the cluster, resulting in arbitrary code execution. This highlights the need for proper configuration management and access control policies in distributed systems (Woodie 2024). Misconfigurations have also affected MongoDB, mainly due to instances left with default settings or insufficient authentication. Such configurations have exposed vast amounts of sensitive data, sometimes involving millions of records. For example, MongoDB databases accessible without authentication have led to large-scale data breaches, including where 275 million records were exposed due to misconfigured settings (Dwivedi et al., 2023). This demonstrates the critical importance of secure configurations in database systems to prevent unauthorized access.

4.3 Third-party Library Risks

Many data science tools rely on external libraries to enhance functionality and reduce development effort. However, these dependencies can introduce vulnerabilities, making systems susceptible to exploitation if the libraries are not sufficiently monitored or updated. One of the most prominent examples of such risks is the Log4j vulnerability, also known as “Log4Shell.” This vulnerability was discovered in Apache Log4j, a widely used Java-based logging library. This flaw allowed attackers to execute arbitrary code remotely by sending specially crafted log messages. The vulnerability impacted multiple data science tools, including Tableau, SAS, RapidMiner, and Databricks, which rely on Log4j for their logging functionality (“Apache Log4j2 vulnerability (Log4shell)” 2022) (What’s New in RapidMiner AI Hub 9.9.3? 2021). In these cases, attackers could exploit the vulnerability to compromise servers, steal sensitive data, or disrupt services. The Log4j incident highlighted the importance of managing third-party libraries effectively.

Tools like Databricks and SAS responded swiftly by releasing patches and updates to address the flaw. Organizations using these tools were also advised to upgrade to secure versions of Log4j and apply additional mitigation measures, such as restricting access to vulnerable components (Administration & Architecture 2020) (Remote Code Execution Vulnerability: CVE-2021-44228 2021). Beyond Log4j, using outdated or unsupported libraries remains a persistent issue. Developers often include third-party libraries for efficiency but neglect to monitor them for security updates. This creates a significant risk if vulnerabilities are discovered in those libraries. For example, poorly maintained Python packages on PyPI or unpatched open-source libraries used by data science platforms can expose systems to attacks (Ruohonen, Hjerppe, and Rindell 2021).

4.4 Remote Code Execution (RCE)

Remote Code Execution (RCE) vulnerabilities pose a severe threat, allowing attackers to execute arbitrary commands on the affected system. These vulnerabilities can compromise systems’ integrity, confidentiality, and availability, enabling attackers to gain unauthorized control or launch further attacks. For instance, Qlik Sense Enterprise encountered multiple RCE vulnerabilities, including CVE-2023-41265 and CVE-2023-48365 (Pearson, Bjerg Jensen, and Adey 2024). These flaws allow attackers to execute malicious shell commands on the server hosting Qlik Sense. Exploiting these vulnerabilities, attackers could escalate privileges, compromise sensitive data, or disrupt operations by deploying unauthorized code. Such incidents illustrate the critical importance of applying timely patches and conducting regular security assessments (Critical

Security fixes for Qlik Sense Enterprise for Windows (CVE-2023-41266, CVE-2023-41265) 2024). Also, Google Colab, a popular cloud-based data science tool, faced an RCE vulnerability. This flaw allowed attackers to execute arbitrary code within the Colab environment, potentially compromising user data, disrupting ongoing work, or exploiting connected resources. Google addressed the issue promptly by deploying patches and strengthening the platform's security measures. However, this case underscores the risks inherent in shared or cloud-based computing environments where vulnerabilities can affect multiple users simultaneously (Cybersecurity risks to artificial intelligence 2024).

5. HIGHLIGHTS OF MAJOR INCIDENTS

The Log4j vulnerability (CVE-2021-44228) affected tools like Tableau, SAS, RapidMiner, and Databricks. This vulnerability exploited a flaw in the logging mechanism, enabling attackers to execute malicious code remotely by crafting specific log messages. The issue arose from the extensive use of the Log4j library in these tools, making it a widespread threat. Vendors responded by releasing patches and providing detailed guidance on mitigating the vulnerability. Organizations were urged to update their Log4j versions to secure releases and closely monitor systems for exploitation signs ("Apache Log4j2 vulnerability (Log4Shell)" 2022) (Remote Code Execution Vulnerability: CVE-2021-44228 2021) (What's New in RapidMiner AI Hub 9.9.3? 2021).

RCE in Tableau November 2023, encountered a significant Remote Code Execution (RCE) vulnerability caused by an issue in the Apache ActiveMQ clients used within its architecture. This flaw allowed attackers with network access to execute arbitrary shell commands, potentially leading to unauthorized system control and data exposure. Tableau addressed the issue by releasing product updates and urging customers to apply the necessary patches promptly. This incident emphasized the critical need for regular patch management and robust monitoring systems (Gupta 2023).

MongoDB Misconfiguration 2019, a major misconfiguration in a MongoDB database led to the exposure of approximately 275 million records. The database was left accessible without proper authentication, allowing any internet user to access sensitive information. This breach highlighted the dangers of inadequate configuration and default settings in database systems. Similarly, Toyota experienced a decade-long data leak in 2023 due to misconfigured NoSQL databases, affecting over two million customer records. These incidents underscore the need for strict authentication measures, regular audits, and secure configuration practices to prevent unauthorized access (Dwivedi et al., 2023).

SQL Injection in SAS, Flawed input sanitization led to vulnerabilities in SAS that allowed attackers to manipulate database queries and access sensitive information. By injecting malicious commands, attackers can gain unauthorized access to sensitive data or modify database content. Organizations using SAS are advised to rigorously validate user inputs and adopt parameterized queries to prevent such attacks. These measures help ensure that user inputs do not compromise the system's security (Yarlagadda and Pydipalli 2018).

Directory Traversal in KNIME (CVE-2022-44749) was discovered in the KNIME Analytics Platform that allowed attackers to overwrite arbitrary files on a user's system by exploiting ZIP archive extraction routines. Such attacks could lead to data corruption, unauthorized file access, or even system compromise if sensitive files were targeted. KNIME responded promptly by releasing updated versions of the affected systems, urging users to upgrade to secure versions. This incident highlights the importance of regularly updating software to protect against emerging vulnerabilities (*Security Advisories* 2024).

Hadoop Misconfiguration in YARN Resource Manager 2024, services allowed unauthorized users to submit arbitrary applications, leading to code execution within the Hadoop cluster. This vulnerability poses significant risks, including unauthorized data access or system disruptions. To mitigate such risks, organizations using Hadoop were advised to implement stricter access controls, properly configure security settings, and regularly audit their deployments (Woodie

2024).

Google Colab RCE Vulnerability 2022, enabled attackers to execute arbitrary code remotely. This flaw, caused by insufficient isolation between user environments, posed risks of compromising user data and affecting the platform's integrity. Google addressed the issue by releasing security patches to enhance sandboxing and isolate user environments. This incident underscores the necessity of robust code isolation and regular vulnerability assessments in cloud-based platforms (Cybersecurity risks to artificial intelligence 2024).

In 2023, the Cactus ransomware group exploited multiple vulnerabilities in **Qlik Sense Enterprise,** including CVE-2023-41266 and CVE-2023-41265. These vulnerabilities involved improper validation of HTTP headers and path traversal, enabling attackers to escalate privileges and execute arbitrary shell commands remotely. By leveraging these flaws, the attackers gained unauthorized access to networks, deployed ransomware, and exfiltrated sensitive data. Qlik released patches to address these vulnerabilities and advised customers to apply updates immediately. This incident highlights the critical importance of timely patch management and proactive security monitoring (Critical Security fixes for Qlik Sense Enterprise for Windows (CVE-2023-41266, CVE-2023-41265) 2024) (Kovacs 2023).

6. COMPARATIVE ANALYSIS AND DISCUSSION

This section highlights the balance between usability and security, a critical factor often overlooked in existing studies. Tools with robust security measures may face challenges due to complexity, whereas user-friendly tools may lack sufficient protection. By addressing this trade-off, we aim to provide understanding for developers, data scientists, and organizations to improve the security of their workflows and tools. The discussion builds on prior research while introducing a novel security ranking methodology that integrates multiple evaluation dimensions, such as compliance with industry standards, ease of use, and mitigation of specific risks. Through this analysis, we identify practical recommendations and opportunities for future research to address existing gaps and create a more secure data science ecosystem.

6.1 Security Rankings

The security ranking (TABLE 1) is determined by assessing each tool's security features and vulnerabilities and provides a comprehensive overview of various data science tools, categorized by type, such as proprietary, open-source, and cloud-based. Each tool is evaluated based on its inherent security features and known vulnerabilities, topping in a security ranking where a lower number indicates a higher security standing. Tools with robust security measures, such as comprehensive encryption, multi-factor authentication, and industry standards compliance, are ranked higher.

Contrarily, tools with significant vulnerabilities, including misconfiguration risks, dependency on third-party services, or susceptibility to code injection, are ranked lower. Cloud-based platforms like GCP and IBM Watson Studio are positioned at the top of the ranking. These platforms offer integrated security features, including Identity and Access Management (IAM), default encryption, and extensive compliance certifications, contributing to their higher security standing.

Therefore, proprietary tools such as TIBCO Spotfire and Qlik are placed in the mid-tier. While they provide strong user authentication, data encryption, and granular access control, potential misconfigurations and the need for regular updates impact their security ranking.

Finally, open-source tools like Jupyter Notebook and Google Colab are ranked lower due to inherent vulnerabilities. Despite offering features like token-based authentication and SSL/TLS encryption, they are susceptible to code execution risks, token exposure, and configuration issues, necessitating diligent security practices.

Tool	Type	Security Features	Vulnerabilities	Security Ranking
Google Cloud Platform (GCP)	Cloud-Based	Integrated security features (IAM, encryption by default, security monitoring),Compliance certifications	Misconfiguration risks, Third-party service vulnerabilities	1
IBM Watson Studio	Cloud-Based	Comprehensive encryption, Multi-factor authentication, Compliance with regulations	Complex configurations, Third-party integration risks	2
Databricks	Cloud-Based	Comprehensive access controls, Data encryption, Network security	Shared infrastructure risks, Complexity in management	3
TIBCO Spotfire	Proprietary	Role-based access control, Data encryption, Integration with enterprise security	Complex deployment, Third-party risks	4
Qlik	Proprietary	Strong user authentication, Data encryption, Granular access control	Potential misconfigurations, Regular up- Dates needed	5
Alteryx	Proprietary	User authentication, Data encryption, Audit logging	Integration risks, Complex configuration	6
Tableau	Proprietary	User authentication, Data encryption, Security updates	High cost, Data sharing security risks	7
SAS	Proprietary	Strong encryption algorithms, Robust access control mechanisms	High cost, Dependency on vendor for security updates	8
MATLAB	Proprietary	Proprietary security features, Built-in Support for data encryption	Dependency on vendor for updates, Integration challenges	9
BigML	Cloud-Based	Dataencryption,Secureuserauthentication,GDPRcompliance	Limited control over cloud data, Integration vulnerabilities	10
KNIME	OpenSource	SSL/TLS encryption, Role-based access control, Audit logging	Requires careful configuration, Code injection risks	11
RapidMiner	OpenSource	Data encryption, Secure user authentication, Fine-grained access control	Security gaps in third-party extensions, Diligent update management needed	12
Apache Spark	OpenSource	Kerberos authentication, SSL encryption, Role-based access control	Misconfigurations, Community- Contributed code vulnerabilities	13
Apache Hadoop	OpenSource	Kerberos authentication, Data encryption and tokenization	Configuration issues, Vulnerability to DDoS and data tampering	14
PostgreSQL	OpenSource	Strong access control, Data encryption (SSL/TLS,TDE), Extensive auditing	SQL injection risks, Complex security settings	15
NoSQL Databases (e.g.,MongoDB, Cassandra)	OpenSource	Authentication and access control, SSL/TLS encryption, Secure backup options	NoSQL injection risks, Configuration vulnerabilities	16
MySQL	OpenSource	User roles and permissions, Data encryption (SSL/TLS), Secure connections	SQL injection risks, Configuration issues	17
Python	OpenSource	Wide range of security libraries and Frameworks (e.g., cryptography, PyJWT)	Susceptible to code injection, Lacks built- In security features	18
R	OpenSource	Secure coding practices,	Slower package updates,	19

		Integration with Secure data storage solutions	Susceptible to Code injection	
Jupyter Notebook	OpenSource	Token-based authentication, Configurable HTTP headers, SSL/TLS encryption, Extension management	Code execution risks, Token exposure, Configuration issues	20
Google Colab	Cloud-Based	Google account authentication, Data encryption (intransitandatrest), Isolated environment	Code execution risks, Session management issues, Dependency on Google	21

TABLE 1: Data Science Tools with Security Features, Vulnerabilities, and Security Ranking.

6.2 Critical Discussion and Comparative Evaluation

This study presents a critical comparative analysis that significantly extends prior data science tool security research. Unlike previous works that primarily focus on specific tools or narrow security aspects (Al-Khateeb and Agarwal 2020; Kuszczynski and Walkowski 2023; Carvalho 2024), our study provides a holistic overview by comparing diverse tools, including cloud-based, proprietary, and open-source platforms.

For instance, prior studies have examined the encryption protocols of individual platforms like Google Cloud but failed to evaluate their integration with access controls and compliance standards comprehensively. Our findings emphasize that tools with integrated security measures, such as GCP and IBM Watson Studio, outperform others due to their ability to address multiple vulnerabilities simultaneously.

Additionally, we highlight new vulnerabilities not emphasized in earlier studies, such as the risks introduced by misconfigurations in widely used open-source tools like Jupyter Notebook and KNIME. These tools rely heavily on user expertise, making them more susceptible to exploitation through improper configurations. The study’s focus on third-party dependencies adds a new dimension, as prior research has largely overlooked the risks posed by unverified third-party libraries.

Furthermore, our comparative analysis shows that usability often conflicts with security. While proprietary tools like Tableau and Qlik provide granular access controls, their usability suffers due to complex configurations, resulting in lower adoption rates among smaller organizations. On the other hand, open-source tools like Python and R prioritize flexibility and ease of use but fall short in built-in security measures, leaving them vulnerable to code injection attacks. This usability-security trade-off underscores the need for designing tools that balance both aspects, a gap explicitly addressed by our study.

6.3 Implications and Future Research Directions

The practical implications of this research are significant. Data scientists and developers can use this study to improve the security configurations of their workflows. Organizations can implement the recommended practices to reduce risks and enhance the safety of sensitive data. Moreover, the findings can guide tool developers in designing better security features for future updates.

Future research should focus on addressing the identified gaps. Studies could explore automated solutions for detecting and fixing misconfigurations. Research can also examine the development of more secure third-party libraries and frameworks. Additionally, further work is needed to understand how to maintain usability while improving security. This includes developing intuitive interfaces that make advanced security mechanisms accessible to users. By tackling these areas, future studies can build on the foundation laid by this research, contributing to a more secure data science ecosystem.

7. CONCLUSION

This survey has examined the security features and vulnerabilities associated with various data science tools, including proprietary software, open-source platforms, and cloud-based solutions. While many of these tools implement advanced security measures like encryption and role-based access control, they remain vulnerable to misconfigurations, code injection attacks, and risks linked to third-party libraries.

Misconfigurations, such as databases with improper authentication or default settings, have resulted in significant data breaches. Vulnerabilities in third-party components, such as the Log4j flaw, have exposed tools to severe security risks. Furthermore, attacks like SQL and NoSQL injection underscore the need for robust input validation and secure coding practices.

Organizations must take proactive measures to secure their data science environments. Regular software updates, proper configuration management, and continuous monitoring are essential to mitigate known vulnerabilities and enhance system security. This research emphasizes these proactive measures and underscores the inclusion of future work to resolve identified gaps, such as developing automated solutions and designing user-friendly security interfaces.

By addressing these challenges, this study provides a foundation for maintaining the security of data science tools and environments, ensuring their reliability and trustworthiness in handling sensitive data.

8. REFERENCES

Ablahd, Ann Zeki (2023). “Using python to detect web application vulnerability”. *Res Militaris* 13.2, pp. 1045-1058.

Acito, Frank (2023). “Predictive analytics with KNIME”. *Analytics for citizen data scientists*. Switzerland: Springer.

Administration & Architecture (2020). https://community.databricks.com.
Alharbi, Fuad S (2020). “Dealing with Data Breaches Amidst Changes In Technology.” *International Journal of Computer Science and Security* (IJCSS) 14.3, pp. 108-115.

Almalki, Sultan Ahmed and Jia Song (2020). “A review on data falsification-based attacks in cooperative intelligent transportation systems”. *International Journal of Computer Science and Security* (IJCSS) 14, p. 22.

Al-khateeb, Samer and Nitin Agarwal (2020). “Social cyber forensics: leveraging open source information and social network analysis to advance cyber security informatics”. *Computational and Mathematical Organization Theory* 26, pp. 412-430.

Ambrosio-Cestero, Gregorio, Jose-Raul Ruiz-Sarmiento, and Javier Gonzalez-Jimenez (2023). “The Robot@ Home2 dataset: A new release with improved usability tools”. *SoftwareX* 23, p. 101490.

Ankam, Venkat (2016). *Big data analytics*. Packt Publishing Ltd.
Antiga, Luca Pietro Giovanni, Eli Stevens, and Thomas Viehmann (2020). *Deep learning with PyTorch*. Simon and Schuster.

Apache Log4j2 vulnerability (Log4shell) (2022).
https://kb.tableau.com/QuickFix?id=kA46Q000000oNkl.

Aziz, Khadija, Dounia Zaidouni, and Mostafa Bellafkih (2018). “Real-time data analysis using Spark and Hadoop”. 2018 4th international conference on optimization and applications (ICOA). IEEE, pp. 1-6.

Baviskar, M. R., Nagargoje, P. N., Deshmukh, P. A., & Baviskar, R. R. (2021). A survey of data science techniques and available tools. *International Research Journal of Engineering and Technology* (IRJET), 8.04, 4258-4263.

Bhosale, Harshawardhan S and Devendra P Gadekar (2014). “A review paper on big data and hadoop”. *International Journal of Scientific and Research Publications* 4.10, pp. 1-7.

Bilokon, Paul, Oleksandr Bilokon, and Saeed Amen (2023). “A compendium of data sources for data science, machine learning, and artificial intelligence”. *arXiv preprint* arXiv:2309.05682.

Boros, Attila Péter, Péter Lehotay-Kéry, and Attila Kiss (2023). “Performance impact of network security features on log processing with spark”. *Annales Universitatis Scientiarum Budapestinensis de Rolando Eotvos Nominatae. Sectio Computatorica*. Vol. 55.

Buratti, B. J., Eichmann, P., Shang, Z., Zgraggen, E., Blanc, J., Bowditch, N., ... & Yang, P. (2023). Should Drag-and-Drop Analytics Become Part of the Data Scientist Toolkit?.

Cao, Phuong (2024). “Jupyter Notebook Attacks Taxonomy: Ransomware, Data Exfiltration, and Security Misconfiguration”. *arXiv preprint* arXiv:2409.19456.

Carvalho, Marcelo de (2024). “A Data Reference Architecture for Brazilian Electrical Companies”. PhD thesis. PUC-Rio.

Cecil, Roy R and Jorge Soares (2019). “IBM Watson studio: a platform to transform data to intelligence”. *Pharmaceutical Supply Chains-Medicines Shortages*, pp. 183-192.

Chandra, K. U., Teja, R. S., Arelli, S., & Das, D. (2022, November). CattleCare: IoT-Based Smart Collar for Automatic Continuous Vital and Activity Monitoring of Cattle. In 2022 International Conference on Futuristic Technologies (INCOFT). IEEE, pp. 1-7.

Chang, Ming-Li Emily and Hui Na Chua (2019). “SQL and NoSQL database comparison: from performance perspective in supporting semi-structured data”. *Advances in Information and Communication Networks: Proceedings of the 2018 Future of Information and Communication Conference* (FICC), Vol. 1. Springer, pp. 294-310.

Ciaburro, Giuseppe (2017). *MATLAB for machine learning*. Packt Publishing Ltd.
Cloud Security Report: Tableau Cloud Security in the Cloud (2024). https://www.tableau.com/learn/whitepapers/tableau-online-security-cloud.

Cyber security risks to artificial intelligence (2024). https://www.gov.uk/government/publications/research-on-the-cyber-security-of-ai/cyber-security-risks-to-artificial-intelligence.

Critical Security fixes for Qlik Sense Enterprise for Windows (CVE-2023-41266, CVE-2023-41265) (2024). https://customerportal.,qlik.com/article/Critical-Security-fixes-for-Qlik-Sense-Enterprise-for-Windows-CVE-2023.

Dwivedi, S., Balaji, R., Ampatt, P., & Sudarsan, S. D. (2023, December). A Survey on Security Threats and Mitigation Strategies for NoSQL Databases: MongoDB as a Use Case. In *International Conference on Information Systems Security*. Cham: Springer Nature Switzerland, pp. 57-76.

Edwards, Dr Jason (2024). “Audit Log Management”. *Critical Security Controls for Effective Cyber Defense: A Comprehensive Guide to CIS 18 Controls*. Springer, pp. 211-245.

Eisenmann, Thomas R (2008). “Managing proprietary and shared platforms”. *California management review* 50.4, pp. 31–53.

Elhalid, Osama Burak, Zaynelabdin Alm Alhelal, and Samer Hassan (2023). “Exploring the

Fundamentals of Python Programming: A comprehensive guide for beginners”. *International Journal of Computer and Information Sciences*.

Elshawi, R., Sakr, S., Talia, D., & Trunfio, P. (2018). Big data systems meet machine learning challenges: towards big data science as a service. *Big data research*, 14, pp. 1-11.

Empowering Organizations with Solutions They Can Trust (n.d.). https://www.qlik.com/us/trust?.
Fillbrunn, A., Dietz, C., Pfeuffer, J., Rahn, R., Landrum, G. A., & Berthold, M. R. (2017). KNIME for reproducible cross-domain analysis of life science data. *Journal of biotechnology*, 261, pp. 149-156.

Fowdur, T. P., Beeharry, Y., Hurbungs, V., Bassoo, V., & Ramnarain-Seetohul, V. (2018). Big data analytics with machine learning tools. *Internet of things and big data analytics toward next-generation intelligence*, pp. 49-97.

Galluccio, E., Caselli, E., & Lombari, G. (2020). *SQL injection strategies: Practical techniques to secure old vulnerabilities against modern attacks*. Packt Publishing Ltd.

Ghaffar, A. (2020). Integration of business intelligence dashboard for enhanced data analytics capabilities.

Gorelik, A. (2019). *The enterprise big data lake: Delivering the promise of big data and data science*. O'Reilly Media.

Gupta, N. (2023). Critical Apache vulnerabilities—Impact of Tableau. https://community.tableau.com/s/question/0D58b0000BgN6AyCQK/critical-apache-vulnerabilities-impact-of-tableau.

Gupta, P. (2021). *Practical data science with Jupyter: Explore data cleaning, pre-processing, data wrangling, feature engineering, and machine learning using Python and Jupyter* (English edition). Bpb Publications.

Gupta, Y. K., & Kumari, S. (2020). A study of big data analytics using Apache Spark with Python and Scala. In *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)* (pp. 471-478). IEEE.

Hajare, R., Hodage, R., Wangwad, O., Mali, Y., & Bagwan, F. (2021). Data security in cloud. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)*, 8(3), 240-245.

Haq, H. B. U., Kayani, H. U. R., Toor, S. K., Zafar, S., & Khalid, I. (2020). The popular tools of data sciences: Benefits, challenges and applications. *IJCSNS*, 20(5), 65.

Hazra, R., Chatterjee, P., Singh, Y., Podder, G., & Das, T. (2024). Data encryption and secure communication protocols. In *Strategies for E-Commerce Data Security: Cloud, Blockchain, AI, and Machine Learning* (pp. 546-570). IGI Global.

Henry, E., Heath, I., & de Jong, P. (2022). Workflow automation in Alteryx for tax season processes.

Hussein, A. A. (2020). Using Hadoop technology to overcome big data problems by choosing proposed cost-efficient scheduler algorithm for heterogeneous Hadoop system (BD3). *Journal of Scientific Research and Reports*, 26(9), 58-84.

Islam, M., Shamsa, K., Khush, B., Khadija, K., Muhammad, U., & Rashid, K. (2020). Data-driven decision support system: A business intelligence approach. *American Journal of Engineering*, 5.

Kimm, H., Paik, I., & Kimm, H. (2021). Performance comparison of TPU, GPU, CPU on Google Colaboratory over distributed deep learning. In *2021 IEEE 14th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC)* (pp. 312-319). IEEE.

Klein, B. T., Tyler, C., & Fields, S. (2022). DevOps and data: Faster-time-to-knowledge through SageOps, MLOps, and DataOps.

Kotu, V., & Deshpande, B. (2014). *Predictive analytics and data mining: Concepts and practice with RapidMiner*. Morgan Kaufmann.

Kovacs, E. (2023). Qlik Sense vulnerabilities exploited in ransomware attacks. https://www.securityweek.com/qlik-sense-vulnerabilities-exploited-in-ransomware-attacks/?.

Kuszczynski, K., & Walkowski, M. (2023). Comparative analysis of open-source tools for conducting static code analysis. *Sensors*, 23(18), 7978.

L'Esteve, R. (2022). Databricks. In *The Azure Data Lakehouse Toolkit: Building and Scaling Data Lakehouses on Azure with Delta Lake, Apache Spark, Databricks, Synapse Analytics, and Snowflake* (pp. 83-139). Springer.

Labbe, P., Anjos, C., Solanki, K., & DiMaso, J. (2019). *Hands-On Business Intelligence with Qlik Sense: Implement self-service data analytics with insights and guidance from Qlik Sense experts*. Packt Publishing Ltd.

Lakshmanan, V. (2022). *Data science on the Google Cloud Platform*. O'Reilly Media.

Lavin, M. (2016). Using Jupyter notebooks to build code literacy and introduce digital humanities.

Llerena, L., Rodriguez, N., Castro, J. W., & Acuña, S. T. (2019). Adapting usability techniques for application in open source software: A multiple case study. *Information and Software Technology, 107*, 48-64.

Mahdy, I. H., Rahman, M., Meem, F. I., & Roy, P. P. (2024). Comparative study between observed and numerical downscaled data of surface air temperature. *World Journal of Advanced Research and Reviews, 23(1)*, 2019-2034.

Marasinghe, M. G., & Koehler, K. J. (2018). *Statistical data analysis using SAS*.

Martinez, I., Viles, E., & Olaizola, I. G. (2021). Data science methodologies: Current challenges and future approaches. *Big Data Research, 24*, 100183.

Martinez, W. L., Martinez, A. R., & Solka, J. (2017). *Exploratory data analysis with MATLAB*. Chapman and Hall/CRC.

Meem, F. I., & Mishu, N. D. R. (2023). An evaluation of machine learning models for deep learning image classification with Fashion-MNIST dataset.

Mendez, K. M., Pritchard, L., Reinke, S. N., & Broadhurst, D. I. (2019). Toward collaborative open data science in metabolomics using Jupyter notebooks and cloud computing. *Metabolomics*, 15, 1-16.

Miller, J. D. (2019). *Hands-On Machine Learning with IBM Watson: Leverage IBM Watson to implement machine learning techniques and algorithms using Python*. Packt Publishing Ltd.

Mishu, N. D. R., Meem, F. I., Ridwan, A. E., Rahman, M. M., & Mary, M. M. (2021). Quantum error correction using quantum convolutional neural network (Thesis). Brac University.

Mittal, M., & Raheja, N. G. (2024). *Data visualization and storytelling with Tableau*. CRC Press.

Molin, S. (2021). *Hands-On Data Analysis with Pandas: A Python data science handbook for data collection, wrangling, analysis, and visualization*. Packt Publishing Ltd.

Morabito, V., & Morabito, V. (2016). Data visualization. In *The Future of Digital Business Innovation: Trends and Practices* (pp. 61-83).

Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., López García, Á., Heredia, I., ... & Hluchý, L. (2019). Machine learning and deep learning frameworks and libraries for large-scale data mining: A survey. *Artificial Intelligence Review, 52*, 77-124.

NoSQL injection (2024). https://portswigger.net/web-security/nosql-injection?.

Ogutu, J. O. (2016). A methodology to test the richness of forensic evidence of database storage engine: Analysis of MySQL update operation in InnoDB and MyISAM storage engines (PhD thesis). University of Nairobi.

Omar, H. K., & Jumaa, A. K. (2019). Big data analysis using Apache Spark MLlib and Hadoop HDFS with Scala and Java. *Kurdistan Journal of Applied Research, 4*(1), 7-14.

Ombiro, Z. B. H. (2016). Mobile–based multi-factor authentication scheme for mobile banking (PhD thesis). University of Nairobi.

Pala, S. K. (2021). Databricks analytics: Empowering data processing, machine learning and real-time analytics. *Machine Learning, 10*(1).

Paper, D. (2019). *Hands-on Scikit-Learn for Machine Learning Applications: Data Science Fundamentals with Python*. Apress.

Parmar, R. R., Roy, S., Bhattacharyya, D., Bandyopadhyay, S. K., & Kim, T. H. (2017). Large-scale encryption in the Hadoop environment: Challenges and solutions. *IEEE Access, 5*, 7156-7163. https://doi.org/10.1109/ACCESS.2017.2694431

Patel, A. (2021). *Data visualization using Tableau*.

Pavlenko, L. V., Pavlenko, M. P., Khomenko, V. H., & Mezhuyev, V. I. (2022). Application of R programming language in learning statistics. *Proceedings of the 1st Symposium on Advances in Educational Technology, 2*, 62-72.

Pearson, E., Jensen, R. B., & Adey, P. (2024). Pred-Pol-Pov: Visibility, data flows, and the predictive policing of poverty. *Surveillance & Society, 22*(2), 120-137. https://doi.org/10.24908/ss.v22i2.7993

Pereira, R. F., Silva, R. M., & Orvalho, J. P. (2020). Virtualization and security aspects: An overview. *International Journal of Computer Science and Security (IJCSS), 14*(5), 154-163.

Pokorný, J. (2020). JSON functionally. In *Advances in Databases and Information Systems: 24th European Conference, ADBIS 2020, Lyon, France, August 25–27, 2020, Proceedings 24* (pp. 139–153). Springer. https://doi.org/10.1007/978-3-030-49992-7_12

Pope, D. (2017). *Big data analytics with SAS: Get actionable insights from your big data using the power of SAS*. Packt Publishing Ltd.

Purgindla, V. R. (2018). *Data processing and the envision*.

Ramasubramanian, K., & Singh, A. (2017). *Machine learning using R* (1st ed.). Springer.

Ramuka, M. (2019). *Data analytics with Google Cloud platform*. BPB Publications.

Ranjan, M. K., Barot, K., Khairnar, V., Rawal, V., Pimpalgaonkar, A., Saxena, S., & Sattar, A. M. (2023). *Python: Empowering data science applications and research*.

Raschka, S., Patterson, J., & Nolet, C. (2020). Machine learning in Python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *Information, 11*(4), 193. https://doi.org/10.3390/info11040193

Rawat, B., & Purnama, S. (2021). MySQL database management system (DBMS) on FTP site LAPAN Bandung. *International Journal of Cyber and IT Service Management, 1*(2), 173-179. https://doi.org/10.1016/j.ijcism.2021.03.005

Reece, M., Lander, T. E., Stoffolano, M., Sampson, A., Dykstra, J., Mittal, S., & Rastogi, N. (2023). Systemic risk and vulnerability analysis of multi-cloud environments. *arXiv preprint* arXiv:2306.01862. https://arxiv.org/abs/2306.01862

Sasikala, V. (2017). Big data analytics steps and tools used in the analytical process. *Journal of Management and Science, 7*(1), 183-195.

Security Advisories. (2024). *KNIME Security Advisories*. https://www.knime.com/security/advisories

Security advisory regarding TIBCO Spotfire. (2023). *TIBCO Support*. https://support.tibco.com/external/article?articleUrl=Security-Advisory-regarding-TIBCO-Spotfire-20231010

Security and compliance guide. (2024). *Databricks Documentation*. https://docs.databricks.com/en/security/index.html

Security guide-Azure Databricks. (2024). *Microsoft Learn*. https://learn.microsoft.com/en-us/azure/databricks/security

Security overview. (2024). *Google Cloud Documentation*. https://cloud.google.com/docs/security

Serra, A. M., Estima, J., & Rodrigues da Silva, A. (2023). Evaluation of Maestro, an extensible general-purpose data gathering and data classification platform. *Information Processing & Management, 60*(5), 103458. https://doi.org/10.1016/j.ipm.2023.103458

Sewal, P., & Singh, H. (2021). A critical analysis of Apache Hadoop and Spark for big data processing. In *2021 6th International Conference on Signal Processing, Computing and Control (ISPCC)* (pp. 308–313). IEEE. https://doi.org/10.1109/ISPCC51984.2021.00061

Shukla, S., George, J. P., Tiwari, K., & Kureethara, J. V. (2022). Data ethics and challenges. In *Data Ethics and Challenges* (pp. 41-59). Springer. https://doi.org/10.1007/978-3-030-65548-1_5

Sial, A. H., Rashdi, S. Y. S., & Khan, A. H. (2021). Comparative analysis of data visualization libraries Matplotlib and Seaborn in Python. *International Journal, 10*(1), 277-281. https://doi.org/10.1016/j.ijcss.2021.03.004

Sohag, S. R., Zhang, S., Xian, M., Sun, S., Xu, F., & Ma, Z. (2024). Causality extraction from nuclear licensee event reports using a hybrid framework. *arXiv preprint* arXiv:2404.05656. https://arxiv.org/abs/2404.05656

Sukhdeve, S. R., & Sukhdeve, S. S. (2023a). Google Colaboratory. In *Google Cloud Platform for Data Science: A Crash Course on Big Data, Machine Learning, and Data Analytics Services* (pp. 11-34). Springer. https://doi.org/10.1007/978-3-030-85843-1_2

Sukhdeve, S. R., & Sukhdeve, S. S. (2023b). Introduction to GCP. In *Google Cloud Platform for Data Science: A Crash Course on Big Data, Machine Learning, and Data Analytics Services* (pp. 1-9). Springer. https://doi.org/10.1007/978-3-030-85843-1_1

Sveen, A. F. (2019). Efficient storage of heterogeneous geospatial data in spatial databases. *Journal of Big Data, 6*(1), 102. https://doi.org/10.1186/s40537-019-0171-6

Svolba, G. (2017). *Applying data science: Business case studies using SAS*. SAS Institute.

Tall, A. M., & Zou, C. C. (2023). A framework for attribute-based access control in processing big data with multiple sensitivities. *Applied Sciences, 13*(2), 1183. https://doi.org/10.3390/app13021183

Tian, T. (2020). Social big data: Techniques and recent applications. *International Journal of Computer Science and Security (IJCSS), 14*(5), 224-233.

Uta, A., Ghit, B., Dave, A., Rellermeyer, J., & Boncz, P. (2022). In-memory indexed caching for distributed data processing. In *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)* (pp. 104-114). IEEE. https://doi.org/10.1109/IPDPS53613.2022.00022

VanderPlas, J. (2016). *Python data science handbook: Essential tools for working with data*. O’Reilly Media, Inc.

Wahid, A., & Kashyap, K. (2019). Cassandra-A distributed database system: An overview. In *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2018, Volume 1* (pp. 519-526). Springer.

Weiss, C. J. (2020). A creative commons textbook for teaching scientific computing to chemistry students with Python and Jupyter notebooks. *Journal of Chemical Education, 98*(2), 489-494. https://doi.org/10.1021/acs.jchemed.0c00277

What’s new in RapidMiner AI Hub 9.9.3? (2021). *RapidMiner Documentation*. https://docs.rapidminer.com/9.10/hub/releases/changes-9.9.3.html

Wickham, H., Çetinkaya-Rundel, M., & Grolemund, G. (2023). *R for data science*. O’Reilly Media, Inc.

Wijaya, A. (2022). *Data engineering with Google Cloud Platform: A practical guide to operationalizing scalable data analytics systems on GCP*. Packt Publishing Ltd.

Woodie, A. (2024). New Hadoop and Flink hacks leveraging known configuration vulnerability. *Datanami*. https://www.datanami.com/2024/01/10/new-hadoop-and-flink-hacks-leveraging-known-configuration-vulnerability/

Xavier, M. (2013). *TIBCO Spotfire for Developers*. Packt Publishing.

Yang, X., Wang, X., Liu, Z., & Shu, F. (2022). M2Coder: A fully automated translator from Matlab M-functions to C/C++ codes for ACS motion controllers. In *International Conference on Guidance, Navigation and Control* (pp. 3130-3139). Springer Nature Singapore. https://doi.org/10.1007/978-3-030-68240-2_376

Yarlagadda, V. K., & Pydipalli, R. (2018). Secure programming with SAS: Mitigating risks and protecting data integrity. *Engineering International, 6*(2), 211–222. https://doi.org/10.26713/engint.051_6.2.1388

Zahra, S., & Ashif, A. (2020). A generic view of big data: Tools and techniques. *International Journal of Computing and Information Science, 1*(1), 1-13.

Zhai, Y., Yin, L., Chase, J., Ristenpart, T., & Swift, M. (2016). CQSTR: Securing cross-tenant applications with cloud containers. In *Proceedings of the Seventh ACM Symposium on Cloud Computing* (pp. 223-236). ACM. https://doi.org/10.1145/2987550.2987569