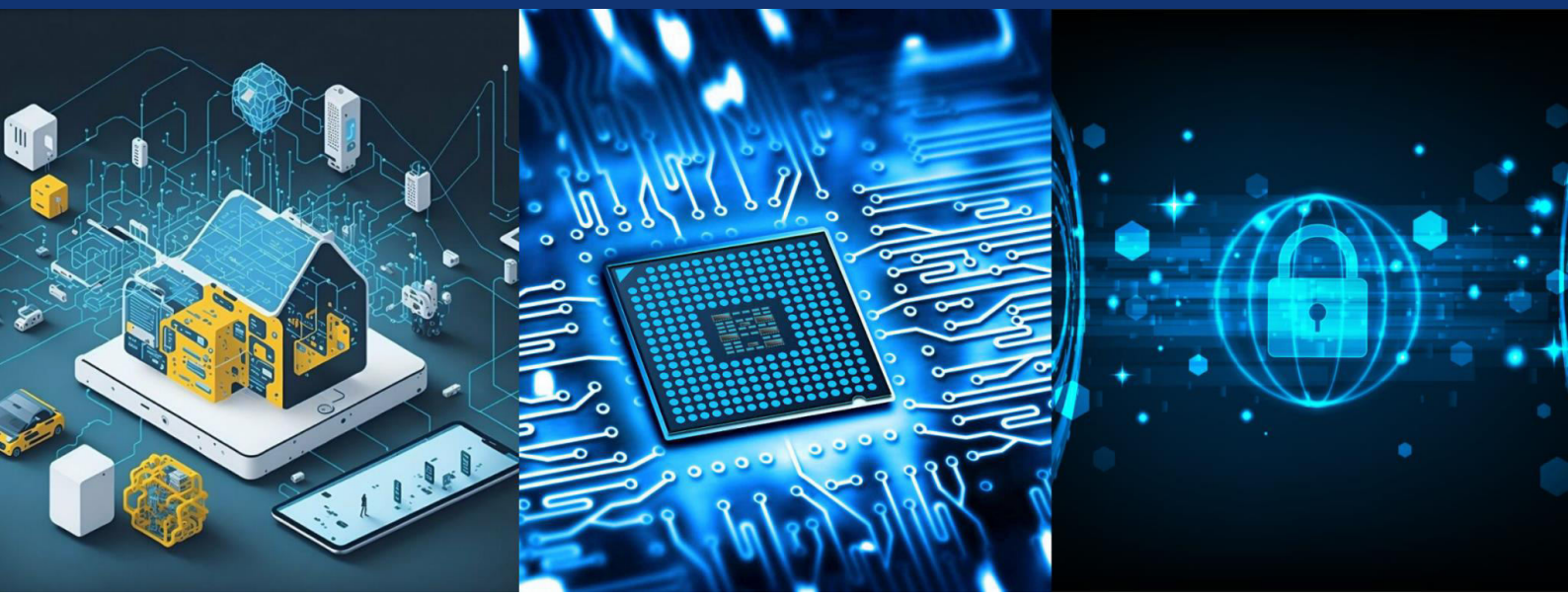
IJIRCCE

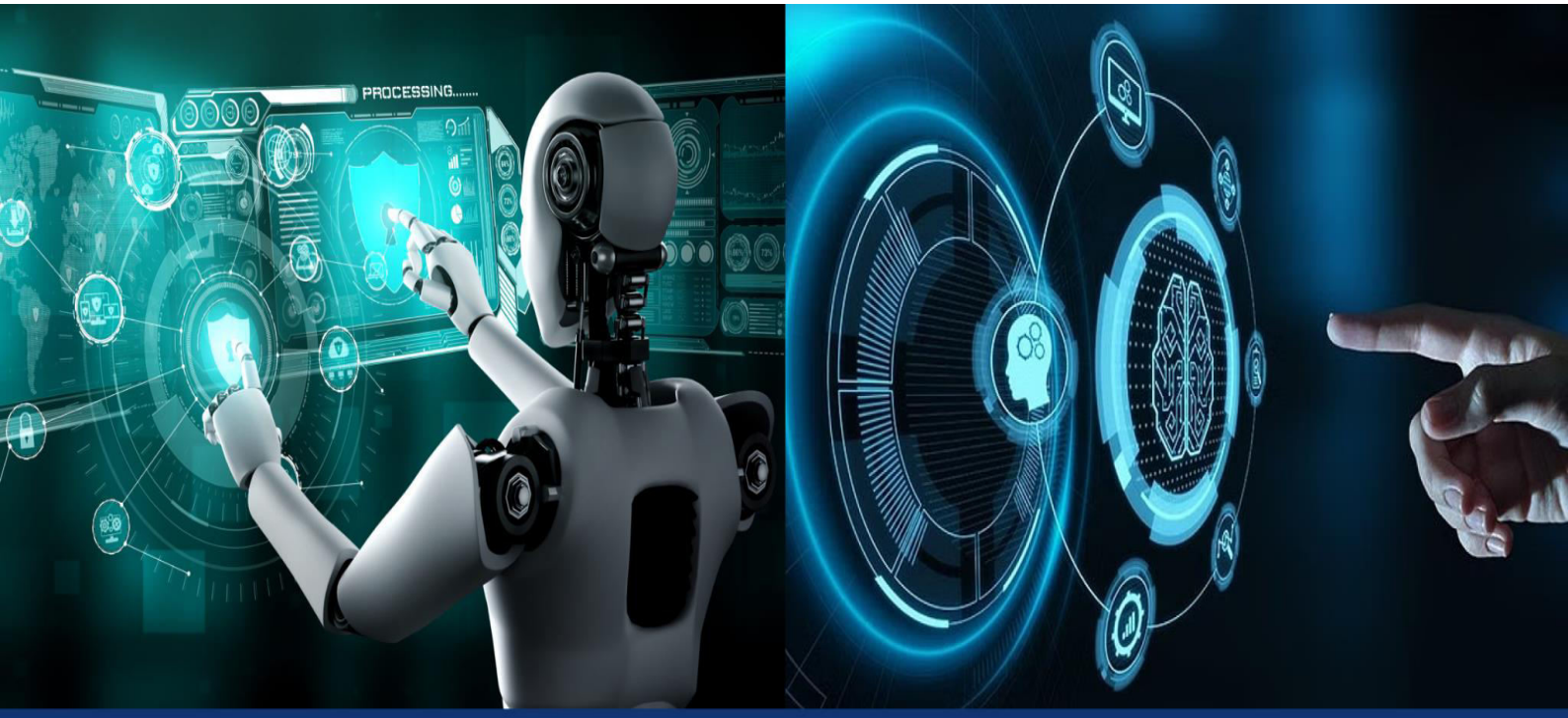
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.771

Volume 13, Issue 5, May 2025

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305037

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Machine Learning-Based URL Safety Detection with Privacy Protection using Cryptography

Dr. C. M. Selvarani, R. Rathish, M. Deepak, K. Lakshmanan

Professor, Department of Cyber Security, Muthayammal Engineering College, Rasipuram, India

Student, Department of Cyber Security, Muthayammal Engineering College, Rasipuram, India

ABSTRACT: Online safety is frequently and seriously at danger from malicious URLs and websites. Naturally, search engines are the cornerstone of information management. However, our users are now seriously at risk due to the widespread presence of bogus websites on search engines. The majority of methods used today to identify rogue websites focus on a specific attack. Online safety is frequently and seriously at danger from malicious URLs and websites. Naturally, search engines are the cornerstone of information management. However, our users are seriously at risk due to the rise of bogus websites on search engines. The majority of methods used today to identify rogue websites focus on a specific attack. However, a lot of websites remain unaffected by the widely accessible blacklist-based browser addons. Any data leaving the client side must be properly disguised, as the server cannot infer any meaningful information from the masked data. Here, the recommended initial Privacy-Preserving Safe Browsing (PPSB) service is given. Robust security assurances are given, which the existing SB services do not offer. The suggested method uses blacklist storage to identify malicious URL access. SVM classification was used to classify the user-provided input URL. SVM is a class of machine learning algorithm that reliably determines the safety or riskiness of a URL. Specifically, it retains the ability to identify malicious URLs while protecting the user's privacy, browsing history, and proprietary data of the blacklist provider (the list of dangerous URLs). This project presented a technique that encrypts critical data to safeguard user privacy from outside analysts and service providers. Furthermore, completely supports the functions of chosen aggregates for analysing user behaviour online and guaranteeing differential privacy. The AES encryption method is used to protect user behaviour data online.

I. INTRODUCTION

Phishing imitates the characteristics and features of emails and makes it look the same as the original one. It appears similar to that of the legitimate source. The user thinks that this email has come from a genuine company or an organisation. This makes the user to forcefully visit the phishing website through the links given in the phishing email. These phishing websites are made to mock the appearance of an original organisation website. The phishers force user to fill up the personal information by giving alarming messages or validate account messages etc. so that they fill up the required information which can be used by them to misuse it. They make the situation such that the user is not left with any other option but to visit their spoofed website.

In the training phase, we should use the labeled data in which there are samples such as phish area and legitimate area. If we do this then classification will not be a problem for detecting the phishing domain. To do a working detection model it is very crucial to use data set in the training phase. We should use samples whose classes are known to us, which means the samples that we label as phishing should be detected only as phishing. Similarly the samples which are labeled as legitimate will be detected as legitimate URL. The dataset to be used for machine learning must actually consist these features.

There so many machine learning algorithms and each algorithm has its own working mechanism which we have already seen in the previous chapter. The existing system uses any one of the suitable machine learning algorithms for the detection of phishing URL and predicts its accuracy. The existing system has good accuracy but it is still not the best as phishing attack is a very crucial; we have to find a best solution to eliminate this. In the currently existing system, only one machine learning algorithm is used to predict the accuracy, using only one algorithm is not a good approach to improve the prediction accuracy.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Malicious SB service provider wants to know whether a user is visiting a particular web page, e.g., some political news. One way to achieve this is that the web browser sends all the visited URLs to a remote server, either in the plaintext, hash value or encrypted format. However, this behavior can be detected by monitoring and analyzing the browser, e.g., using the taint analysis technique.

A malicious party might leverage PPSB (Privacy Preserving Safe Browsing) to degrade the client-side user experience, like inserting a number of fake or safe URLs or increasing the server-side delay. To address this potential issue, PPSB provides a flexible mechanism for users to add or remove blacklist providers. Admin could add the fake URL and keyword to this blacklist storage. User can also allowed suggesting the malicious website details regarding black list.

II. SYSTEM ANALYSIS AND EXISTING SYSTEM

In existing system developed client-side defence mechanism based on machine learning techniques to detect spoofed web pages and protect users from phishing attacks. As a proof of concept, a Google Chrome extension dubbed as Phish Catcher, is developed that implements our machine learning algorithm that classifies a URL as suspicious or trustful. The algorithm takes four different types of web features as input and then random forest classifier decides whether a login web page is spoofed or not. Traditional classifiers used techniques like whitelisting, blacklisting, online learning strategies, lexical and host-based analysis of URLs. Blacklisting alone is not efficient as it does not anticipate the status of prior non-visited URLs. Moreover, classifiers based on online strategies were not accurate, while whitelisting and lexical based models had high latency. After web page feature extraction, a random forest classifier model is selected on the basis of the performance metrics such as latency, accuracy and efficiency. Subsequently, the classifier was trained using the supervised machine learning technique. The extracted features were then fed to the selected model in order to complete the learning process. After the completion of the learning process, the model is ready for testing.

Contrary to the traditional approaches, this scheme offers to run the classification in the browser itself. The user interface of our plug-in is made simple for the better understanding of the user. When a user enters a phished URL, it displays a phishing alert on the screen and highlights the corresponding phishing features of that URL in a drop-down menu. The feature-set contains thirty features which are categorized into four groups where each group is acknowledged as a decision tree. Random forest classifier employs the aggregated outcome of the decision trees to identify the bogus and genuine login web pages.

Limitations

- Noise in the training labels, such as misclassified instances, can negatively impact the model's accuracy.
- Imbalanced dataset can lead to reduced accuracy in detecting the minority class.
- Collecting and processing data on client devices for machine learning can raise privacy concerns.
- Incorrectly blocking legitimate access to websites can frustrate users and damage trust.

III. PROPOSED SYSTEM

A malicious party might leverage fake websites to degrade the client-side user experience, like inserting a number of fake or safe URLs or increasing the server-side delay. To address this potential issue, safe browsing website provides a flexible mechanism for users to add or remove blacklist providers. Admin could add the fake URL and keyword to this blacklist storage. User can also allowed suggesting the malicious website details regarding black list. In this system malware detection system uses a supervised machine learning approach for discovering malwares. The SVM based malware detection system extends the idea of signature based detection system with a combination of behavior monitoring approach. It utilizes static and dynamic analysis of malwares by taking the run time traces of the executable. This model also provides search data security which encrypts the users' sensitive data to prevent privacy from both outside analysts and the aggregation service provider. Also, completely supports selective aggregate functions for online user behaviour analysis and guaranteeing differential privacy.

Expected Merits

- Using AES encryption algorithm makes sure that this model is strong secured.
- There is no clue for the server or malicious user to predict the users' online usage of websites.
- Prevent users from accessing malicious websites.

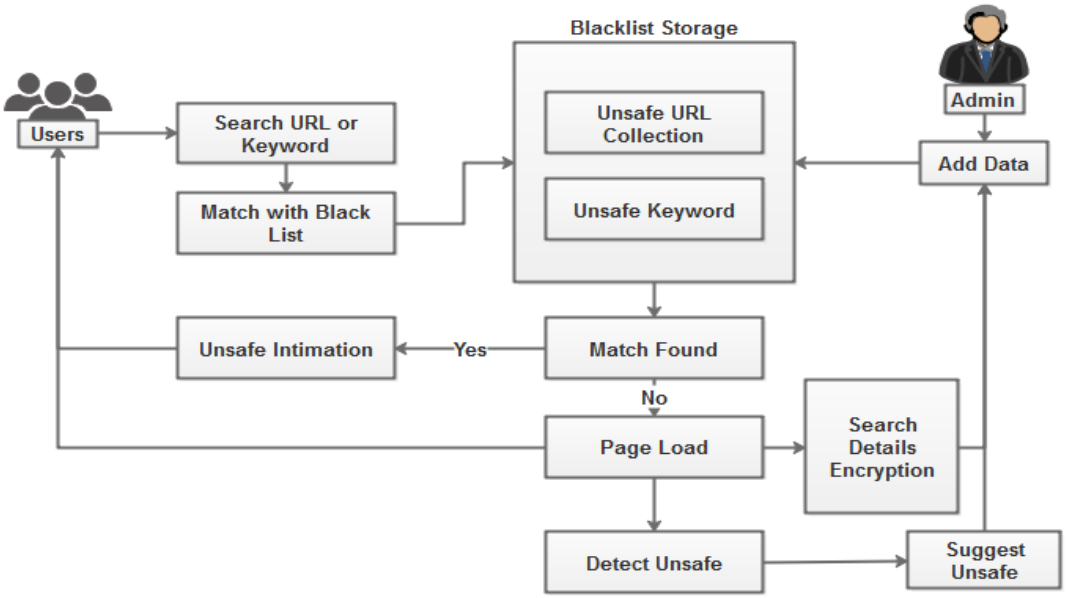
© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305037

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

System Architecture



IV. SOFTWARE DESCRIPTION

4.3.1 PYTHON

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. In July 2018, Van Rossum stepped down as the leader in the language community. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of Python's other implementations. Python and CPython are managed by the non-profit Python Software Foundation. Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach. While offering choice in coding methodology, the Python philosophy rejects exuberant syntax (such as that of Perl) in favor of a simpler, less-cluttered grammar. As Alex Martelli put it: "To describe something as 'clever' is not considered a compliment in the Python culture. "Python's philosophy rejects the Perl "there is more than one way to do it" approach to language design in favour of "there should be one—and preferably only one—obvious way to do it".

Python's developers strive to avoid premature optimization, and reject patches to non-critical parts of C Python that would offer marginal increases in speed at the cost of clarity.[When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. CPython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter. An important goal of Python's developers is keeping it fun to use. This is reflected in the language's name a tribute to the British comedy group Monty Python and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a famous Monty Python sketch) instead of the standard for and bar.

A common neologism in the Python community is pythonic, which can have a wide range of meanings related to program style. To say that code is pythonic is to say that it uses Python idioms well, that it is natural or shows fluency

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305037

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

in the language, that it conforms with Python's minimalist philosophy and emphasis on readability. In contrast, code that is difficult to understand or reads like a rough transcription from another programming language is called un pythonic. Users and admirers of Python, especially those considered knowledgeable or experienced, are often referred to as Pythonists, Pythonistas, and Pythoneers. Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Python's initial development was spearheaded by Guido van Rossum in the late 1980s. Today, it is developed by the Python Software Foundation. Because Python is a multiparadigm language, Python programmers can accomplish their tasks using different styles of programming: object oriented, imperative, functional or reflective. Python can be used in Web development, numeric programming, game development, serial port access and more.

There are two attributes that make development time in Python faster than in other programming languages:

1. Python is an interpreted language, which precludes the need to compile code before executing a program because Python does the compilation in the background. Because Python is a high-level programming language, it abstracts many sophisticated details from the programming code. Python focuses so much on this abstraction that its code can be understood by most novice programmers.
2. Python code tends to be shorter than comparable codes. Although Python offers fast development times, it lags slightly in terms of execution time. Compared to fully compiling languages like C and C++, Python programs execute slower. Of course, with the processing speeds of computers these days, the speed differences are usually only observed in benchmarking tests, not in real-world operations. In most cases, Python is already included in Linux distributions and Mac OS X machines.

4.3.2 MYSQL

MySQL is the world's most used open source relational database management system (RDBMS) as of 2008 that run as a server providing multi-user access to a number of databases. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack—LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, Word Press, phpBB, MyBB, Drupal and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google(though not for searches), ImagebookTwitter, Flickr, Nokia.com, and YouTube.

Inter images

MySQL is primarily an RDBMS and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data,

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.

Graphical

The official MySQL Workbench is a free integrated environment developed by MySQL AB, that enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, MySQL GUI Tools. Similar to other third-party packages, but still considered the authoritative MySQL frontend, MySQL Workbench lets users manage database design & modeling, SQL development (replacing MySQL Query Browser) and Database administration (replacing MySQL Administrator).MySQL Workbench is available in two editions, the regular free and open source Community Edition which may be downloaded from the MySQL website, and the proprietary Standard Edition which extends and improves the feature set of the Community Edition.

4.3.3 PYCHARM

PyCharm is an integrated development environment (IDE) for Python programming language, developed by JetBrains. PyCharm provides features such as code completion, debugging, code analysis, refactoring, version control integration, and more to help developers write, test, and debug their Python code efficiently. PyCharm is available in two editions: Community Edition (CE) and Professional Edition (PE). The Community Edition is a free, open-source version of the IDE that provides basic functionality for Python development. The Professional Edition is a paid version of the IDE that provides advanced features such as remote development, web development, scientific tools, database tools, and more. PyCharm is available for Windows, macOS, and Linux operating systems. It supports Python versions 2.7, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, and 3.10.

Features:

- Intelligent code completion
- Syntax highlighting
- Code inspection
- Code navigation and search
- Debugging
- Testing
- Version control integration
- Web development support
- Scientific tools support
- Database tools support

Integration with other JetBrains tools

PyCharm's code completion feature can help speed up development by automatically suggesting code based on context and previously written code. It also includes a debugger that allows developers to step through code, set breakpoints, and inspect variables. PyCharm has integration with version control systems like Git, Mercurial, and Subversion. It also supports virtual environments, which allow developers to manage different Python installations and packages in isolated environments. The IDE also has features specifically geared towards web development, such as support for popular web frameworks like Django, Flask, and Pyramid. It includes tools for debugging, testing, and profiling web applications. PyCharm also provides scientific tools for data analysis, visualization, and scientific computing, such as support for NumPy, SciPy, and matplotlib. It also includes tools for working with databases, such as PostgreSQL, MySQL, and Oracle. Overall, PyCharm is a powerful and feature-rich IDE that can greatly increase productivity for Python developers.

Customization:

PyCharm allows developers to customize the IDE to their liking. Users can change the color scheme, fonts, and other settings to make the IDE more comfortable to use. PyCharm also supports plugins, which allow developers to extend the IDE with additional features.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Collaboration:
makes it easy for developers to collaborate on projects. It supports integration with popular collaboration tools such as GitHub, Bitbucket, and GitLab. It also includes features for code reviews, task management, and team communication.

Education:
PyCharm provides a learning environment for Python programming language. PyCharm Edu is a free, open-source edition of PyCharm that includes interactive courses and tutorials for learning Python. It provides an easy-to-use interface for beginners and includes features such as code highlighting, autocompletion, and error highlighting.

Support:
PyCharm has an active community of users who provide support through forums and social media. JetBrains also provides comprehensive documentation, tutorials, and training courses for PyCharm. For users who need more personalized support, JetBrains offers a paid support plan that includes email and phone support.

Pricing:
PyCharm Community Edition is free and open-source. PyCharm Professional Edition requires a paid license, but offers a 30-day free trial. JetBrains also offers a subscription-based pricing model that includes access to all JetBrains IDEs and tools.

Integrations:
PyCharm integrates with a wide range of tools and technologies commonly used in Python development. It supports popular Python web frameworks like Flask, Django, Pyramid, and web2py. It also integrates with tools for scientific computing like NumPy, SciPy, and pandas. PyCharm also supports popular front-end technologies such as HTML, CSS, and JavaScript.

Performance:
PyCharm is known for its fast and reliable performance. It uses a combination of static analysis, incremental compilation, and intelligent caching to provide fast code completion and navigation. PyCharm also has a memory profiler that helps identify and optimize memory usage in Python applications.

Ease of Use:
PyCharm provides an intuitive and easy-to-use interface for developers. It has a well-organized menu structure, clear icons, and easy-to-navigate tabs. PyCharm also provides a variety of keyboard shortcuts and customizable keymaps that allow users to work efficiently without constantly switching between the mouse and keyboard.

V. EXPECTED OUCTOME

In this proposed work, implement a Malicious URL Detection process using machine learning techniques. This focuses on detecting unsafe website URLs and keywords with the help of encrypted blacklist storage. According to few selected features can be used to differentiate between legitimate and malicious web pages. These selected features are many such as URLs and Keywords. In proposed work a service provider that owns a high-quality blacklist, which may be more frequently updated or simply contains more items. User also allowed to directly sharing blacklists with servers in an uncontrollable way could make these dataset be obtained by every user. With the help of efficient classification approach will detect the fake websites accurately and prevent the users from accessing that websites. This also provides the secure encryption approach avoid the unknown access of search history. The security is provided to the search data which has been stored in the database

VI. FEATURES

Malicious URL Detection

- Identifies unsafe website URLs using machine learning techniques.
- Differentiates between legitimate and malicious web pages based on key features like URLs and keywords.

Encrypted Blacklist Storage

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

- Stores blacklists in encrypted form to enhance security.
- Ensures frequent updates and high-quality blacklist maintenance.

Blacklist Sharing Mechanism
- Allows controlled sharing of blacklists with servers.
- Prevents unauthorized access to the dataset while enabling efficient updates.

Efficient Classification Approach
- Uses machine learning models to accurately detect and block fake websites.
- Prevents users from accessing harmful websites.

Search History Protection
- Implements secure encryption to protect search history.
- Ensures only authorized access to stored search data.

Database Security
- Provides robust security for stored search data.
- Safeguards user information from unauthorized access.

REFERENCES

[1] Sonowal, Gunikhan, and K. S. Kuppusamy. "PhiDMA–A phishing detection model with multi-filter approach." Journal of King Saud University-Computer and Information Sciences 32, no. 1 (2020): 99-112.
[2] Alaparthi, Shivaji, and Manit Mishra. "Bidirectional Encoder Representations from Transformers (BERT): A sentiment analysis odyssey." arXiv preprint arXiv:2007.01127 (2020).
[3] Ni, Jianjun, Yu Cai, Guangyi Tang, and Yingjuan Xie. "Collaborative filtering recommendation algorithm based on TF-IDF and user characteristics." Applied Sciences 11, no. 20 (2021): 9554.
[4] Ahammad, SK Hasane, Sunil D. Kale, Gopal D. Upadhye, Sandeep Dwarkanath Pande, E. Venkatesh Babu, Amol V. Dhumane, and Mr Dilip Kumar Jang Bahadur. "Phishing URL detection using machine learning methods." Advances in Engineering Software 173 (2022): 103288.
[5] Mourtaji, Youness, Mohammed Bouhorma, Daniyal Alghazzawi, Ghadah Aldabbagh, and Abdullah Alghamdi. "Hybrid rule-based solution for phishing URL detection using convolutional neural network." Wireless Communications and Mobile Computing 2021 (2021): 1-24.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details