

- [Research Journals](https://www.gyanvihar.org/researchjournals/) (<https://www.gyanvihar.org/researchjournals/>)

SOFTWARE VULNERABILITY DEPENDENT ON MACHINE LEARNING

[Home](#) / [Volume 6 Issue 2, 2020](#) / [Software Vulnerability Dependent On Machine Learning](#)

SOFTWARE VULNERABILITY DEPENDENT ON MACHINE LEARNING

 VOLUME 6 ISSUE 2, 2020

([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/CATEGORY/INTERNATIONAL-JOURNAL-OF-CONVERGING-TECHNOLOGIES-MANAGEMENT/VOLUME-6-ISSUE-2-2020-INTERNATIONAL-JOURNAL-OF-CONVERGING-TECHNOLOGIES-MANAGEMENT/](https://www.gyanvihar.org/journals/index.php/category/international-journal-of-converging-technologies-management/volume-6-issue-2-2020-international-journal-of-converging-technologies-management/))

Markad Ashok Vitthalrao

Research Scholar

Dr. Mukesh Kumar Gupta

Professor

ashok.markad@gmail.com (mailto:ashok.markad@gmail.com)

mukeshkr.gupta@mygyanvihar.com

Computer Engineering, School of Engineering & Technology,

Suresh Gyan Vihar University, Mahal, Jagatpura

Abstract

Software vulnerabilities are main reasons of various security attacks, so it will greatly compromise the safety of the system, and may even cause losses. With the undeniably rich of weakness related information and the broad use of machine learning strategies, programming weakness examination strategies dependent on AI is turning into an significant research region of data security. In this paper, the modern and surely understood works in this investigate territory is discussed. We have thought about, the restrictions of these works.

Introduction:

Programming security framework includes an enormous number of complex issues. While numerous product improvement associations set forward a progression of programming improvement security standards and best practices to improve the security of programming frameworks. Be that as it may, the developing of programming framework vulnerabilities has turned out to be one of the principle dangers to security of data frameworks.

To lessen the security imperfections in programming frameworks, programming defenselessness investigation is turning into the focal point of data framework security innovation examine. After twenty years of advancement, the principle research comprise of static defenselessness investigation, dynamic helplessness examination furthermore, blend of them. Security apparatuses have been connected to all phases of programming improvement to lessen the harms brought about by programming security issues, including configuration, coding, testing, arrangement, and so forth.. Static defenselessness examination strategy has a preferred position in speed, yet, it for the most part has a high rate of false positives; dynamic defenselessness examination strategy can precisely find the vulnerabilities, yet for huge scale programming frameworks, this strategy more often than not require an excess of assets. Blend of static and dynamic weakness examination strategy has better identification productivity and precision, yet it more often than not suits a specific helplessness type.

AI can consequently produce information through a lot of information, and after that utilizing the learning for forecast. It has connected in the field of content arrangement and vindictive code discovery. As the expanding information of programming powerlessness, it has turned into an significant job to utilize AI in programming powerlessness investigation. This paper focuses on a product weakness examination system dependent on machine learning and arranges and portrays the current techniques. At long last, the constraint of current weakness examination technique dependent on machine learning is talked about.

AI strategies are connected effectively in content arrangement and pernicious code discovery field. In content characterization fields, Salton vector space model is utilized to express record as an accumulation of words, and after that insert them into vector space, AI techniques are utilized

to concentrate highlights to produce characterization. In the discovery of vindictive code, composing explicit location marks that can match recognize contaminated has on the organize, AI techniques can essentially improve the recognition exactness.

During the preparation arrange, the greater security program code and the powerless program code are preparing set, after the program investigation and highlight extraction, the consequence of these two stages were contribution to AI calculations, (for example, choice trees, neural systems, and so forth.) what's more, got the classifier of programming powerlessness investigation, appeared in *Figure 1a*. In the investigation organize, the program code was broke down and highlight extraction right off the bat, through the classifier to decide whether it contains the weakness, and find the weakness position as per the attributes of weakness, appeared in *Figure 1b*.

From the viewpoint of program analysis the existing techniques can be separated into lexical examination, syntactic investigation and semantic examination, typically syntactic investigation

incorporates lexical investigation; From the point of view of highlights extraction, the techniques can be separated into security code include extraction and weakness code highlight extraction; from the viewpoint of powerlessness information, the techniques can be isolated into realized weakness designs what's more, obscure vulnerabilities designs.

Existing vulnerability analysis method based on machine learning:

The current helplessness examination strategies dependent on AI are typically identified with three unique sorts of grouping strategies. Program examination is the establishment of highlight extraction and AI, Be that as it may, the greatness of the program examination, separating highlights and AI technique have greater contrasts. In this segment, we will depict existing strategies from the point of view of program examination.

1. vulnerability analysis based on lexical analysis:

Yamaguchi Fabian et al. [5] proposed a strategy utilizing AI to distinguish vulnerabilities of source codes in 2011. Right off the bat, this strategy recognizes API images of each work through lexical examination. It inserts API images in vector space, and measurement information through the head

part examination strategy to discover the use of API mode. At that point, from the viewpoint of the known helplessness capacities, it finds the API utilization designs with surmised capacities, managing code audit to recognize conceivable vulnerabilities. Through the investigation of FFmpeg library, 20 capacities like CVE-2010-3429 were found from 6778 capacities and two of them were affirmed as helplessness, one is a known powerlessness, the other is multi day weakness.

Mokhov and Serguei [16] present a quick machine learning way to deal with static code investigation and fingerprinting for shortcomings identified with security, programming building.

They use N-Gram calculation to investigation paired code as signal. To help discovery of powerless or defenseless code, counting source or twofold on various stages the AI approach demonstrated to be quick and exact to for such undertakings where different instruments are either much slower

or then again have a lot littler review of known vulnerabilities.

In 2012, Fabian improved his methodology [6] on the premise of [5]. Fabian et al. propose a technique for helping a security examiner during inspecting of source code. Their technique continues by removing dynamic sentence structure trees from the code and deciding auxiliary examples in these trees, to such an extent that each capacity in the code can be portrayed as a

blend of these examples. This portrayal empowers them to decay a known weakness and extrapolate it to a code base, with the end goal that capacities possibly experiencing a similar imperfection can be proposed to the investigator. At long last they assess their strategy on the source code of four famous open-source ventures: LibTIFF, FFmpeg, Pidgin and Reference bullet. For three of these undertakings, they can recognize zero-day vulnerabilities by examining just a little part of the code bases.

Cheng Hong et al. propose to display programming executions with charts at two degrees of granularity: strategies and fundamental squares [13]. An individual hub speaks to a strategy or

essential square and an edge speaks to a strategy call, technique return or change (at the strategy or fundamental square granularity). Given a lot of charts of right and flawed executions, we propose to separate the most discriminative sub-charts which difference the program stream of right and broken executions. The separated sub-charts not just pinpoint the bug, yet in addition give an instructive setting for comprehension and fixing the bug. They further broaden it to create a positioned rundown of top-k discriminative sub-charts speaking to particular areas which may contain bugs.

Neuhaus et al. present Vulture instrument [15] that can naturally mine current helplessness databases and adaptation files to outline vulnerabilities to parts. The subsequent positioning of the most powerless parts is an ideal base for further examinations on what makes segments helpless. In an examination of the Mozilla defenselessness history, they shockingly discovered that segments that had a solitary helplessness in the past were by and large not prone to have further vulnerabilities.

Nonetheless, parts that had comparable imports or capacity calls were probably going to be helpless.

Almorsy et al. present another arrangement that supports mechanized weakness investigation utilizing formalized helplessness marks [17]. Rather than relying upon formal strategies to find weakness cases where analyzers must be created to find explicit vulnerabilities, their methodology fuses a formal helplessness mark portrayed utilizing OCL. Utilizing this formal mark, they perform program examination of the target framework to find mark matches (for example indications of potential vulnerabilities). A newfound helplessness can be effectively recognized in an objective program gave that a formal mark for it exists. They have built up a model

static weakness examination device dependent on our formalized weakness marks particular methodology. They have approved our methodology in catching marks of the OWSAP Top10 vulnerabilities and connected these marks in investigating a lot of seven benchmark applications.

Medeiros et al. investigate the utilization of a crossover of strategies to recognize vulnerabilities with less false positives [18]. After an beginning advance that utilizations spoil examination to hail applicant vulnerabilities, their methodology utilizes information mining to foresee the presence of false positives. This methodology comes to a exchange off between two clearly inverse methodologies: people coding the information about vulnerabilities (for spoil examination) versus naturally acquiring that information (with AI, for information mining). Given this progressively exact type of identification, they do programmed code revision by embeddings fixes in the source code. The approach was executed in the WAP device and an test assessment was performed with an enormous arrangement of open source PHP applications.

Shar and Lwin Khin give an elective answer for existing corrupt analyzers by proposing static code properties [19] that can be utilized to anticipate explicit program explanations, as opposed to programming segments, which are liable to be powerless against SQLI or XSS. From the perceptions of info disinfection code that are normally executed in web applications to stay away from SQLI and XSS vulnerabilities. In this paper, they propose a lot of static code properties that describe such code designs. They at that point assemble defenselessness forecast models from the

chronicled data that reflect proposed static qualities furthermore, known defenselessness information to foresee SQLI and XSS vulnerabilities. At last, they built up a model instrument

called PhpMinerI for information gathering and utilized it to assess their models on eight open source web applications. their best model accomplished an arrived at the midpoint of consequence of 93% review and 11% false caution rate in foreseeing SQLI vulnerabilities, and 78% review and 6% false caution rate in foreseeing XSS vulnerabilities.

1. vulnerability analysis based on semantic analysis:

Fabian et al. present Chucky [7], a strategy to uncover missing checks in source code. Their strategy uses corrupting source code and recognizing abnormal or missing conditions connected to security-basic items.

In 2014, Fabian et al. present a technique to adequately mine a lot of source code for vulnerabilities [8]. They have presented a novel portrayal code property diagram that unions ideas of exemplary program investigation, in particular dynamic language structure trees, control stream charts and program reliance charts, into a joint information structure. This exhaustive portrayal empowers them to exquisitely display formats for normal vulnerabilities with diagram traversals that, for example, can distinguish cushion floods, whole number floods, position string vulnerabilities, or memory exposures. At last, they execute their methodology utilizing a well known chart database what's more, show its adequacy by recognizing 18 beforehand obscure vulnerabilities in the source code of the Linux part.

A technique for naturally deducing quest designs for corrupt style vulnerabilities in C code is proposed by Fabian et al. [9]. Given a security-delicate sink, for example, a memory work, their strategy consequently distinguishes comparing source-sink frameworks and builds designs that model the information stream and cleansing in these frameworks. The deduced examples are communicated as traversals in a code property chart and empower proficiently looking for unsanitized information streams over a few capacities just as with venture explicit APIs. They show the viability of this methodology in various investigations with 5 open source ventures. The surmised pursuit examples lessen the measure of code to assess for finding known vulnerabilities

Grieco et al. propose a defenselessness expectation approach in light of AI [11]. By utilizing fluffing approach to screen the parallel program progressively, Grieco et al. extricate the memory-clashing highlights. They order the information arrangements produced by the dynamic execution to

help AI. They use VDISCOVER to foresee regardless of whether the vulnerabilities happen in the given test suite. They progressively screen 1039 projects in Debian from bug tracker, gathering 138308 execution arrangements and dissecting 76083 diverse capacity call factually. Results demonstrate that some memory spill vulnerabilities are recognized effectively, exhibiting their methodology is successful.

Conclusion

In this paper, we survey the outstanding works that utilization AI advancements to dissect the product vulnerabilities. We previously proposed a product defenselessness examination system dependent on AI. At that point, we classify the most prominent AI innovations into three sorts, depicting them in detail and looking at them obviously. As indicated by the difficulties of utilizing machine learning advances in programming defenselessness investigation, we talk about certain techniques that can mitigate the difficulties.

References

- [1] Wu Shizhong, et al "software vulnerability analysis technology progress." Journal of Tsinghua University (Natural Science) 10 (2012): 1309-1319
- [2] Sebastiani, Fabrizio. "Machine learning in automated text categorization." ACM computing surveys (CSUR) 34.1 (2002): 1-47. APA
- [3] Balzarotti, Davide, et al. "Saner: Composing static and dynamic analysis to validate sanitization in web applications." Security and Privacy, 2008. SP 2008. IEEE Symposium on. IEEE, 2008.
- [4] Yamaguchi, Fabian, Felix Lindner, and Konrad Rieck. "Vulnerability extrapolation: Assisted discovery of vulnerabilities using machine learning." Proceedings of the 5th USENIX conference on Offensive technologies. USENIX Association, 2011.
- [5] Shabtai, Asaf, et al. "Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey." Information Security Technical Report 14.1 (2009): 16-29. APA




- [6] Yamaguchi, Fabian, Markus Lottmann, and Konrad Rieck. "Generalized vulnerability extrapolation using abstract syntax trees." Proceedings of the 28th Annual Computer Security Applications Conference. ACM, 2012.
- [7] Yamaguchi, Fabian, et al. "Chucky: exposing missing checks in source code for vulnerability discovery." Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. ACM, 2013.
- [8] Yamaguchi, Fabian, et al. "Modeling and discovering vulnerabilities with code property graphs." Security and Privacy (SP), 2014 IEEE Symposium on. IEEE, 2014.
- [9] Yamaguchi, Fabian, et al. "Automatic Inference of Search Patterns for Taint-Style Vulnerabilities." (2015).
- [10] Zhang, Su, Doina Caragea, and Xinming Ou. "An empirical study on using the national vulnerability database to predict software vulnerabilities." Database and Expert Systems Applications. Springer Berlin Heidelberg, 2011.
- [11] Grieco, Gustavo, et al. "Toward large-scale vulnerability discovery using Machine Learning." [12] Wang, Yanyan, Yanning Wang, and Jiadong Ren. "Software Vulnerabilities Detection Using Rapid Density-based Clustering." Journal of Computational Information Systems 8.14 (2011): 3295-3302.
- [13] Cheng, Hong, et al. "Identifying bug signatures using discriminative graph mining." Proceedings of the eighteenth international symposium on Software testing and analysis. ACM, 2009.
- [14] Wijayasekara, Dumidu, et al. "Mining bug databases for unidentified software vulnerabilities." Human System Interactions (HSI), 2012 5th International Conference on. IEEE, 2012.
- [15] Neuhaus, Stephan, et al. "Predicting vulnerable software components." Proceedings of the 14th ACM conference on Computer and communications security. ACM, 2007.
- [16] Mokhov, Serguei, Joey Paquet, and Mourad Debbabi. "MARFCAT: Fast code analysis for defects and vulnerabilities." Software Analytics (SWAN), 2015 IEEE 1st International Workshop on. IEEE, 2015.
- [17] Almorsy, Mohamed, John Grundy, and Amani S. Ibrahim. "Supporting automated vulnerability analysis using formalized vulnerability signatures." Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering. ACM, 2012.

SHARE :

f ([http://www.facebook.com/sharer.php?](http://www.facebook.com/sharer.php?u=https://www.gyanvihar.org/journals/index.php/2021/04/23/software-vulnerability-dependent-on-machine-learning/&t=Software%20Vulnerability%20Dependent%20on%20Machine%20Learning)

[u=https://www.gyanvihar.org/journals/index.php/2021/04/23/software-vulnerability-dependent-on-machine-learning/&t=Software Vulnerability Dependent on Machine Learning](https://www.gyanvihar.org/journals/index.php/2021/04/23/software-vulnerability-dependent-on-machine-learning/&t=Software%20Vulnerability%20Dependent%20on%20Machine%20Learning)) **t** ([http://twitter.com/home/?](http://twitter.com/home/?status=Software%20Vulnerability%20Dependent%20on%20Machine%20Learning)

[status=Software Vulnerability Dependent on Machine Learning -](https://www.gyanvihar.org/journals/index.php/2021/04/23/software-vulnerability-dependent-on-machine-learning/)

<https://www.gyanvihar.org/journals/index.php/2021/04/23/software-vulnerability-dependent-on-machine-learning/>)   

Type here to search

**RECENT POSTS**

STUDY OF HYBRID FRP COMPOSITE MADE OF FLAX AND HEMP FIBERS WITH KEVLAR
([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/05/17/STUDY-OF-HYBRID-FRP-COMPOSITE-MADE-OF-FLAX-AND-HEMP-FIBERS-WITH-KEVLAR-2/](https://www.gyanvihar.org/journals/index.php/2021/05/17/study-of-hybrid-frp-composite-made-of-flax-and-hemp-fibers-with-kevlar-2/))

DOCKERIZING VIRTUAL WIRELESS SENSOR NETWORK(VWSN):ISSUES AND SOLUTIONS
([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/05/17/DOCKERIZING-VIRTUAL-WIRELESS-SENSOR-NETWORKVWSNISSUES-AND-SOLUTIONS/](https://www.gyanvihar.org/journals/index.php/2021/05/17/dockerizing-virtual-wireless-sensor-networkvwsnissues-and-solutions/))

CLUSTER-BASED CLASSIFICATION TO PROCESS IMBALANCED DATA USING SMOTE TECHNIQUE
([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/05/17/CLUSTER-BASED-CLASSIFICATION-TO-PROCESS-IMBALANCED-DATA-USING-SMOTE-TECHNIQUE-2/](https://www.gyanvihar.org/journals/index.php/2021/05/17/cluster-based-classification-to-process-imbalanced-data-using-smote-technique-2/))

ULTRAWIDE BAND ANTENNAS BANDWIDTH INCREASING TECHNIQUE
([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/05/17/ULTRAWIDE-BAND-ANTENNAS-BANDWIDTH-INCREASING-TECHNIQUE/](https://www.gyanvihar.org/journals/index.php/2021/05/17/ultrawide-band-antennas-bandwidth-increasing-technique/))

DISCRETE WAVELET TRANSFORM AND DECISION RULES SUPPORTED METHOD FOR DETECTION OF FAULTY EVENTS IN DISTRIBUTION NETWORK WITH RENEWABLE ENERGY
([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/05/17/DISCRETE-WAVELET-TRANSFORM-AND-DECISION-RULES-SUPPORTED-METHOD-FOR-DETECTION-OF-FAULTY-EVENTS-IN-DISTRIBUTION-NETWORK-WITH-RENEWABLE-ENERGY/](https://www.gyanvihar.org/journals/index.php/2021/05/17/discrete-wavelet-transform-and-decision-rules-supported-method-for-detection-of-faulty-events-in-distribution-network-with-renewable-energy/))

COMPARING DIFFERENT MODELS FOR CREDIT CARD FRAUD DETECTION
([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/04/23/COMPARING-DIFFERENT-MODELS-FOR-CREDIT-CARD-FRAUD-DETECTION/](https://www.gyanvihar.org/journals/index.php/2021/04/23/comparing-different-models-for-credit-card-fraud-detection/))

STATIC LOAD ANALYSIS OF SIDE UNDERRUN PROTECTION DEVICE FOR HEAVY COMMERCIAL VEHICLES USING HYPERMESH TOOL
([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/04/23/STATIC-LOAD-ANALYSIS-OF-SIDE-UNDERRUN-PROTECTION-DEVICE-FOR-HEAVY-COMMERCIAL-VEHICLES-USING-HYPERMESH-TOOL/](https://www.gyanvihar.org/journals/index.php/2021/04/23/static-load-analysis-of-side-underrun-protection-device-for-heavy-commercial-vehicles-using-hypermesh-tool/))

COMPARATIVE STUDY OF EXISTING DOCKER AUDIT TOOLS FOR PERFORMING VULNERABILITY DISCOVERY AND AUDITING TASKS IN DOCKER.

([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/04/23/COMPARATIVE-STUDY-OF-EXISTING-DOCKER-AUDIT-TOOLS-FOR-PERFORMING-VULNERABILITY-DISCOVERY-AND-AUDITING-TASKS-IN-DOCKER/](https://www.gyanvihar.org/journals/index.php/2021/04/23/comparative-study-of-existing-docker-audit-tools-for-performing-vulnerability-discovery-and-auditing-tasks-in-docker/))

SOFTWARE VULNERABILITY DEPENDENT ON MACHINE LEARNING

([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/04/23/SOFTWARE-VULNERABILITY-DEPENDENT-ON-MACHINE-LEARNING/](https://www.gyanvihar.org/journals/index.php/2021/04/23/software-vulnerability-dependent-on-machine-learning/))

ANALYSIS OF SURFACE FINISHING PROCESSES-A REVIEW

([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/04/23/ANALYSIS-OF-SURFACE-FINISHING-PROCESSES-A-REVIEW/](https://www.gyanvihar.org/journals/index.php/2021/04/23/analysis-of-surface-finishing-processes-a-review/))

REVIEW PAPER ON HOW TO RECOVER NODE FAILURE USING (MULTI-PROTOCOL LABEL SWITCHING (MPLS) NETWORK AND ENHANCE THE ENERGY OF SIGNAL

([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/02/05/REVIEW-PAPER-ON-HOW-TO-RECOVER-NODE-FAILURE-USING-MULTI-PROTOCOL-LABEL-SWITCHING-MPLS-NETWORK-AND-ENHANCE-THE-ENERGY-OF-SIGNAL/](https://www.gyanvihar.org/journals/index.php/2021/02/05/review-paper-on-how-to-recover-node-failure-using-multi-protocol-label-switching-mpls-network-and-enhance-the-energy-of-signal/))

TO EXPLORE THE ENERGY OF WIRELESS SENSOR NETWORKS TO SOLVE BOTTLENECK PROBLEM USING SWARM ROUTING APPROACH: A REVIEW

([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/02/05/TO-EXPLORE-THE-ENERGY-OF-WIRELESS-SENSOR-NETWORKS-TO-SOLVE-BOTTLENECK-PROBLEM-USING-SWARM-ROUTING-APPROACH-A-REVIEW/](https://www.gyanvihar.org/journals/index.php/2021/02/05/to-explore-the-energy-of-wireless-sensor-networks-to-solve-bottleneck-problem-using-swarm-routing-approach-a-review/))

SECURITY AND PRIVACY CHALLENGES IN INTERNET OF EVERYTHING (IOE) WITH SECURITY REQUIREMENTS

([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/01/29/SECURITY-AND-PRIVACY-CHALLENGES-IN-INTERNET-OF-EVERYTHING-IOE-WITH-SECURITY-REQUIREMENTS/](https://www.gyanvihar.org/journals/index.php/2021/01/29/security-and-privacy-challenges-in-internet-of-everything-ioe-with-security-requirements/))

THE IMPACT OF EXTREMOPHILIC ENZYME: WIDELY APPLICATION IN INDUSTRIAL USES

([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/01/24/THE-IMPACT-OF-EXTREMOPHILIC-ENZYME-WIDELY-APPLICATION-IN-INDUSTRIAL-USES/](https://www.gyanvihar.org/journals/index.php/2021/01/24/the-impact-of-extremophilic-enzyme-widely-application-in-industrial-uses/))

EVALUATION OF PHYSICO-CHEMICAL PARAMETERS OF DRINKING WATER AT VARIOUS SITES OF KOTA, RAJASTHAN

([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/01/24/EVALUATION-OF-PHYSICO-CHEMICAL-PARAMETERS-OF-DRINKING-WATER-AT-VARIOUS-SITES-OF-KOTA-RAJASTHAN/](https://www.gyanvihar.org/journals/index.php/2021/01/24/evaluation-of-physico-chemical-parameters-of-drinking-water-at-various-sites-of-kota-rajasthan/))

PHARMACOLOGICAL POTENTIAL OF CURCUMA CAESIA

([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/01/24/PHARMACOLOGICAL-POTENTIAL-OF-CURCUMA-CAESIA/](https://www.gyanvihar.org/journals/index.php/2021/01/24/pharmacological-potential-of-curcuma-caesia/))

PLANT-PARASITIC NEMATODES AND THEIR MANAGEMENT

([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/01/24/PLANT-PARASITIC-NEMATODES-AND-THEIR-MANAGEMENT/](https://www.gyanvihar.org/journals/index.php/2021/01/24/plant-parasitic-nematodes-and-their-management/))

ANALYTICAL AND KINETIC INVESTIGATION OF AZO DYE ARSENAZO(III) AS COPPER(II) DERIVATIVE BY USING SPECTROSCOPIC METHODS

([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/01/24/ANALYTICAL-AND-KINETIC-INVESTIGATION-OF-AZO-DYE-ARSENAZOIII-AS-COPPERII-DERIVATIVE-BY-USING-SPECTROSCOPIC-METHODS/](https://www.gyanvihar.org/journals/index.php/2021/01/24/analytical-and-kinetic-investigation-of-azo-dye-arsenazoiii-as-copperii-derivative-by-using-spectroscopic-methods/))

SOURCES AND BIOLOGICAL ACTIVITY OF COUMARINS: AN APPRAISAL

([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/01/24/SOURCES-AND-BIOLOGICAL-ACTIVITY-OF-COUMARINS-AN-APPRAISAL/](https://www.gyanvihar.org/journals/index.php/2021/01/24/sources-and-biological-activity-of-coumarins-an-appraisal/))

PRE- AND POST-MONSOON VARIATION IN PHYSICO-CHEMICAL CHARACTERISTICS IN GROUNDWATER QUALITY OF AMER TEHSIL IN JAIPUR, RAJASTHAN

([HTTPS://WWW.GYANVIHAR.ORG/JOURNALS/INDEX.PHP/2021/01/23/PRE-AND-POST-MONSOON-VARIATION-IN-PHYSICO-CHEMICAL-CHARACTERISTICS-IN-GROUNDWATER-QUALITY-OF-AMER-TEHSIL-IN-JAIPUR-RAJASTHAN/](https://www.gyanvihar.org/journals/index.php/2021/01/23/pre-and-post-monsoon-variation-in-physico-chemical-characteristics-in-groundwater-quality-of-amer-tehsil-in-jipur-rajasthan/))



([HTTP://WWW.GYANVIHAR.ORG/RESEARCHJOURNALS/ENGG.PHP](http://www.gyanvihar.org/researchjournals/engg.php))

ISSN : 2394-9570

SURESH GYAN VIHAR UNIVERSITY
INTERNATIONAL JOURNAL OF
ENVIRONMENT
SCIENCE AND
TECHNOLOGY



([HTTP://WWW.GYANVIHAR.ORG/RESEARCHJOURNALS/ENVIRMENTAL_SCIENCE.PHP](http://www.gyanvihar.org/researchjournals/envirmental_science.php))

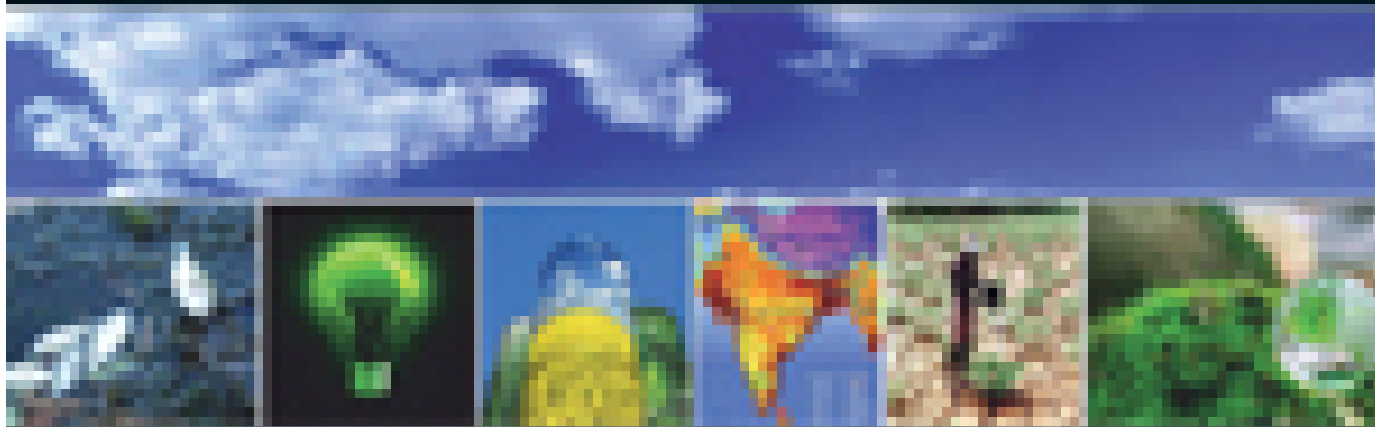


([HTTP://WWW.GYANVIHAR.ORG/RESEARCHJOURNALS/CTM_JOURNALS.PHP](http://www.gyanvihar.org/researchjournals/CTM_JOURNALS.PHP))

ISSN 2250-7798

Suresh Gyan Vihar University

Journal of Climate Change and Water



VOLUME 12 | ISSUE 1 | JANUARY 2021



SURESH
GYAN VIHAR
UNIVERSITY

([HTTP://WWW.GYANVIHAR.ORG/RESEARCHJOURNALS/INT_JOU_CLIMATE_CHANGE_WATER.P](http://www.gyanvihar.org/researchjournals/int_jou_climate_change_water.p)



([HTTP://WWW.GYANVIHAR.ORG/RESEARCHJOURNALS/ECO_MANG_JOUR.PHP](http://www.gyanvihar.org/researchjournals/eco_mang_jour.php))

ISSN-XXXX-XXXX

**SURESH GYAN VIHAR
UNIVERSITY**

**JOURNAL OF PHARMACEUTICAL
RESEARCH & EDUCATION**

VOLUME-I | ISSUE-I | JULY 2016



**SURESH
GYAN VIHAR
UNIVERSITY**

([HTTP://WWW.GYANVIHAR.ORG/RESEARCHJOURNALS/PHARMACY.PHP](http://www.gyanvihar.org/researchjournals/pharmacy.php))

COPYRIGHT © 2018 BY TEAM SGVU. ALL RIGHTS RESERVED.