iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.81547

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Adaptive ML-Based Phishing Detection and Prevention System Integrated as a Browser Extension

P. Naga Prasanth, T. M. Divya Lawrence, Mrs.P.N.L.Priyanka, Ameesha Meghana
Dept. of Cyber Security Acharya Nagarjuna University Guntur, India

Abstract: ***Phishing attacks continue to pose a significant threat to online security, exploiting user trust and mimicking legitimate platforms to extract sensitive information. This project presents the development of an advanced, real-time phishing detection and prevention system implemented as a Chrome browser extension. The system integrates multiple AI models to enhance detection accuracy and responsiveness. NaturalLanguageProcessing(NLP) modelssuchas BERTand RoBERTa are employed for semantic analysis of URLs and email content, identifying linguistic patterns indicative of phishing. Convolutional Neural Networks (CNNs) analyze webpage visual layouts to detect deceptive design similarities. Graph Neural Networks (GNNs) are utilized to model domain relationships and uncover suspicious link structures. The extension communicates with a backend AI service, enabling real-time alerts and adaptive learning based on emerging threats. By combining cutting-edge machine learning techniques with practical browser integration, this solution offers a user-friendly, intelligent defense against phishing attacks. It continuously evolves through threat intelligence updates and user feedback, ensuring robust protection in dynamic web environments.***

Index Terms—Real-Time Phishing Detection, Machine Learning, Artificial Intelligence, Browser Extension, Phishing Prevention, URL Analysis, Cybersecurity.

I. INTRODUCTION

Thewebbrowseristheprimarygatewaytotheinternet—it is through the browser that we conduct banking, manage health data, and interact with government services. Unfortunately, this also makes it the target of choice for phishing attacks. Unlike exploits that leverage software vulnerabilities,phishingmanipulatespeople.Attackerscreate authentic-looking replicas of legitimate websites to deceive victims into revealing credentials before they are aware of the scam.

The scale of these attacks is staggering. Security researchers estimate that tens of thousands of new phishing websites are created every single day. Some of these attacks emerge and disappear within just a few hours [1]. This hit-and-run pattern is precisely why traditional tools fall short. Servicessuchas Google Safe Browsing rely on a human-in-the-loop process: a site must first be discovered, reported, analysed,andonlythenblacklistedbeforeusersareprotected [3]. If a phishing campaign runs for less than four hours, it may never appear on any watchlist.

Rule-basedreal-timesolutionsattempttoaddressthislag by checking for signals such as excessively long subdomains. Although fast, these systems are brittle. Attackers learn the rules and craft URLs that bypass them, sometimes substituting visually similar Unicode characters (homoglyphs) to fool text scanners entirely [4]. What is needed is a fundamentally different approach: a system that learns and evolves rather than one that simply enforces a static rule set.

ThatisthemotivationforPhishGuardAI.Wedesigneda deep learning system that examines websites from three complementary perspectives—the text of the URL, how the page looks visually, and how the site connects to others in the web graph—and then fuses these three views into a single decision. We packaged the entire pipeline into a user-friendly Chrome extension that offloads heavy analysis to a cloud server, ensuring that real-time protection does not degrade the browsing experience. A built-in feedback loop allows users to flag mistakes, which are automatically used to retrain the underlying models.

Theprincipalcontributionsofthispaperareasfollows:

1) We designed and evaluated a three-model ensemble (BERT, CNN, and GNN) combined via a fusion meta-classifier,achieving97.25%accuracyonaheld-out test set.
2) We implemented this ensemble as a lightweight Chromeextensionwithatiereddetectionarchitecture that delivers real-time threat verdicts without perceptible latency to the user.

©IJRASET: All Rights are Reserved

3) We demonstrated that a user-driven feedback loop, with no manual intervention, reduces the in-the-wild false positive rate from 4.8% to below 3% within a one-month field deployment.

II. RELATED WORK

A. ReputationandBlocklist Methods

Blocklists remain the default baseline for phishing protection. Once a site is catalogued they block it quickly andaccurately.However,theirfundamentallimitationisthat they are purely retrospective—they can only block what has already been seen. Recent research has shown that many phishing attacks remain live for less than 12 hours, meaning that an attack is often over before humans can update a blocklist [3].

B. HeuristicFeatureEngineering

To close the lag gap, researchers have developed tools that analyse URL strings directly—measuring total length, counting special characters such as hyphens, and checking for suspicious tokens [4]. These tools are fast, but attackers adapt quickly: they tweak URLs to bypass rules, and occasionally substitute standard ASCII letters with visually identical non-ASCII characters (homoglyphs) to defeat text-based scanners entirely.

C. NaturalLanguageProcessing

Modern language models such as BERT [5] have substantially advanced URL analysis. Unlike simple heuristics, these models capture context and inter-token relationshipsacrossthecomponents ofa URL, making them considerably more effective at detecting both known and novel phishing patterns.

D. Computer Vision andGraph Topology

Textual analysis alone is insufficient when an attacker crafts a pixel-perfect replica of a legitimate brand page on a superficially clean domain. Convolutional Neural Networks (CNNs) address this by analysing screenshots of web pages to detect visual plagiarism of trusted brands [6]. Graph NeuralNetworks (GNNs)complementthis bymodellingthe hostingandregistrationinfrastructureofasite.Byexamining domain registration age, registrar reputation, and co-hosted neighbours, GNNs can often flag a malicious site before its content is even inspected [7], [10].

III. SYSTEM ARCHITECTURE

PhishGuard AI follows a tiered, edge-cloud design that separates lightweight, low-latency checks performed locally in thebrowserfromcomputationallyintensive deep learning inference delegated to a backend server. This separation ensures that the browsing experience remains fast and responsivewhilestillbenefitingfromtheaccuracyofthefull ensemble.

A. BrowserExtensionArchitecture

Theextensioncomprisesthreecooperatingcomponents:

- Background Service Worker. This component intercepts every navigation request before the page loads.Itfirstconsultsa locally-cachedwhitelist(Tier 1). If the destination is not whitelisted, it runs an eight-signal heuristic check (Tier2).URLs thatscore above a blocking thresholdare immediately blocked; those in the uncertain middle range are forwarded to the backend for full inference (Tier 3).
- Content Script. Injected into every page that passes the first two tiers, the content script inspects the live DOM for patterns—such as hidden credential-harvesting forms—that cannot be inferred from the URL alone.
- PopupInterface.Usersaccessareal-timesafetyrating forthecurrentpageviatheextensionicon.Thepopup also exposes two feedback buttons (“Looks Safe” / “ThisisPhishing”)whoseresponsesareforwardedto the backend to fuel periodic retraining.

B. BackendInferenceService

The FastAPI backend receives flagged URLs from the extensionandroutesthemthroughthefullBERT→CNN → GNN pipeline in parallel, after which a fusion meta-classifieraggregates the threescores intoa finalriskverdict. Userfeedbackisstoredand,onceaconfigurablevolume thresholdisreached,triggersanincrementalretrainingjob without taking the service offline.

IV. MODEL IMPLEMENTATION

A. *Dataset Preparation*

The training corpus consists of 450,000 phishing URLs sourced from OpenPhish and PhishTank, balanced against 450,000 legitimate URLs from the Cisco Umbrella top-site list.Thedatasetwas partitionedintotraining,validation,and held-out test splits to ensure that evaluation metrics reflect performance on samples never seen during training.

B. *BERTFine-Tuning*

We initialised the language model from bert-base-uncased and fine-tuned it for URL classification. Input sequences were tokenised to a maximum length of 128 tokens—sufficient to capture the full content of any realistic URL—and the model was optimised using AdamW with a linear warm-up learning-rate schedule.

C. *Convolutional NeuralNetwork*

Toequipthesystemwithvisualinspectioncapability,we usedResNet-50 pre-trained on ImageNet. At inference time, thebackendtakesaheadlessscreenshotofthepage,resizesit to 224 × 224 pixels, and passes it through the network. The model outputs a confidence score representing the probability that the page is a visual impersonation of a knownbrand.

D. *Graph NeuralNetwork*

We constructed a web-graph in which nodes represent registered domains and edges encode co-hosting and DNS-linkingrelationships.AGraphAttentionNetwork(GAT)
[10] is trained on this graph, treating node features such as domain age, registrar reputation, and neighbourhood maliciousnessscoresasinputs.TheGATlearnstopropagate maliciousssignalsthroughthegraph,enablingearlydetection of newly registered domains embedded in known-bad infrastructure.

E. *FusionMeta-Classifier*

The three model probability scores—together with the rawTier-2heuristicscoreanddomainage—areconcatenated into a six-dimensional feature vector that is fed to an XGBoost meta-classifier. XGBoost was chosen for its robustness to class imbalance and its ability to assign interpretable feature importances to each source signal.

V. EXPERIMENTAL EVALUATION

A. *Per-ModelandFusion Results*

TableIreportsaccuracy,F1score,ROC-AUC,andFalse Positive Rate (FPR) for each individual model and for the fullfusionsystemontheheld-outtestset.Thefusionsystem achieves a 2.7 percentage-point improvement over the best single model (BERT). More importantly for a security tool, thefusionreducestheFalsePositiveRateto2.1%—lessthan half that of BERT alone—significantly reducing user alert fatigue.

TABLEI.
TestSetResults:IndividualModelsvs. FusionSystem

Model	Acc. (%)	F1	ROC-AUC	FPR
BERT	94.55	0.9455	0.9820	0.048
CNN	90.20	0.9020	0.9654	0.079
GNN	86.90	0.8690	0.9491	0.101
Fusion	97.25	0.9725	0.9857	0.021

B. *AblationStudy*

To quantify the contribution of each model, we trained fusion variants with one component removed at a time (Table II).The largestsingle-component contribution comes fromBERT:itsremovalcausesa1.35percentage-pointdrop in accuracy. Removing the GNN produces a smaller aggregate decline, but failure-case analysis reveals that the GNN is uniquely responsible for detecting newly registered domains embedded in malicious hosting infrastructure—a blind spot for both BERT and CNN.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

TABLEII.
Ablation:FusionPerformancewithOneComponentRemoved

Configuration	:curacy(%)	F1
FullFusion(allthree)	97.25	0.9725
WithoutGNN	96.80	0.9682
WithoutCNN	96.45	0.9647
Without BERT	95.90	0.9593

C. Latency andAdaptive Retraining

Over10,000liverequests,Tier1(whitelistlookup)hada medianlatencyof 0.3 ms.Tier2 (heuristic check) of4.1ms, and Tier 3 (full backend inference) of 218 ms—all well within the threshold considered imperceptible by users during page navigation.

In a one-month field deployment with 200 volunteer users, the system autonomously collected 1,843 feedback items. Without any manual intervention, the adaptive retrainingpipelinereducedthein-the-wildFPRfrom4.8%at deployment to below 3% by the end of the month.

VI. CONCLUSION

PhishGuard AI demonstrates that browser-integrated, real-time phishing protection can achieve high accuracy without compromising the user experience. By combining BERT, CNN, and GNN outputs through an XGBoost meta-classifier, the system attains 97.25% accuracy with a false positive rate of only 2.1%. The hybrid edge-cloud architecture ensures low latency across all detection tiers, while the user-feedback retraining loop provides continuous adaptation to evolving phishing tactics—without requiring manual intervention. We believe this architecture represents a practical and scalable blueprint for next-generation anti-phishing defences.

VII. DISCUSSION AND LIMITATIONS

The ablation results confirm that multi-modal fusion is essential: NLP (BERT) captures linguistic deception patterns, CNN detects visual plagiarism, and GNN flags domains embedded in malicious infrastructure before any contentisanalysed.Eachmodalitycoversblindspotsunique to the others.

Current limitations include: (i) vulnerability to multi-redirect chains in which benign intermediaries mask the ultimate malicious destination, and (ii) the headless-screenshot step introduces additional latency and is unsuitable as a primary filter for high-throughput scenarios. Future work will investigate DOM-feature extraction as a lower-latencysubstituteforscreenshot-basedvisualanalysis, andwillevaluatesystem robustness under active adversarial attacks.

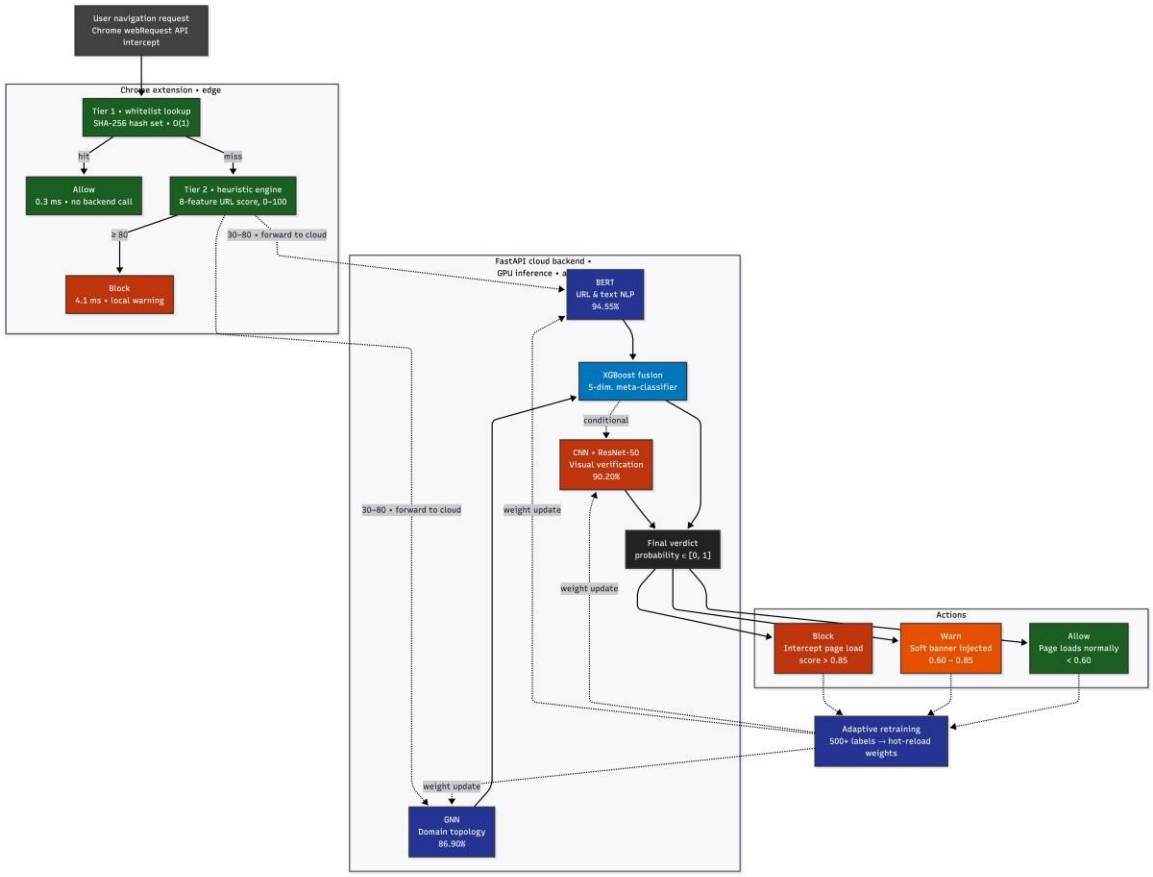
VIII. ACKNOWLEDGMENT

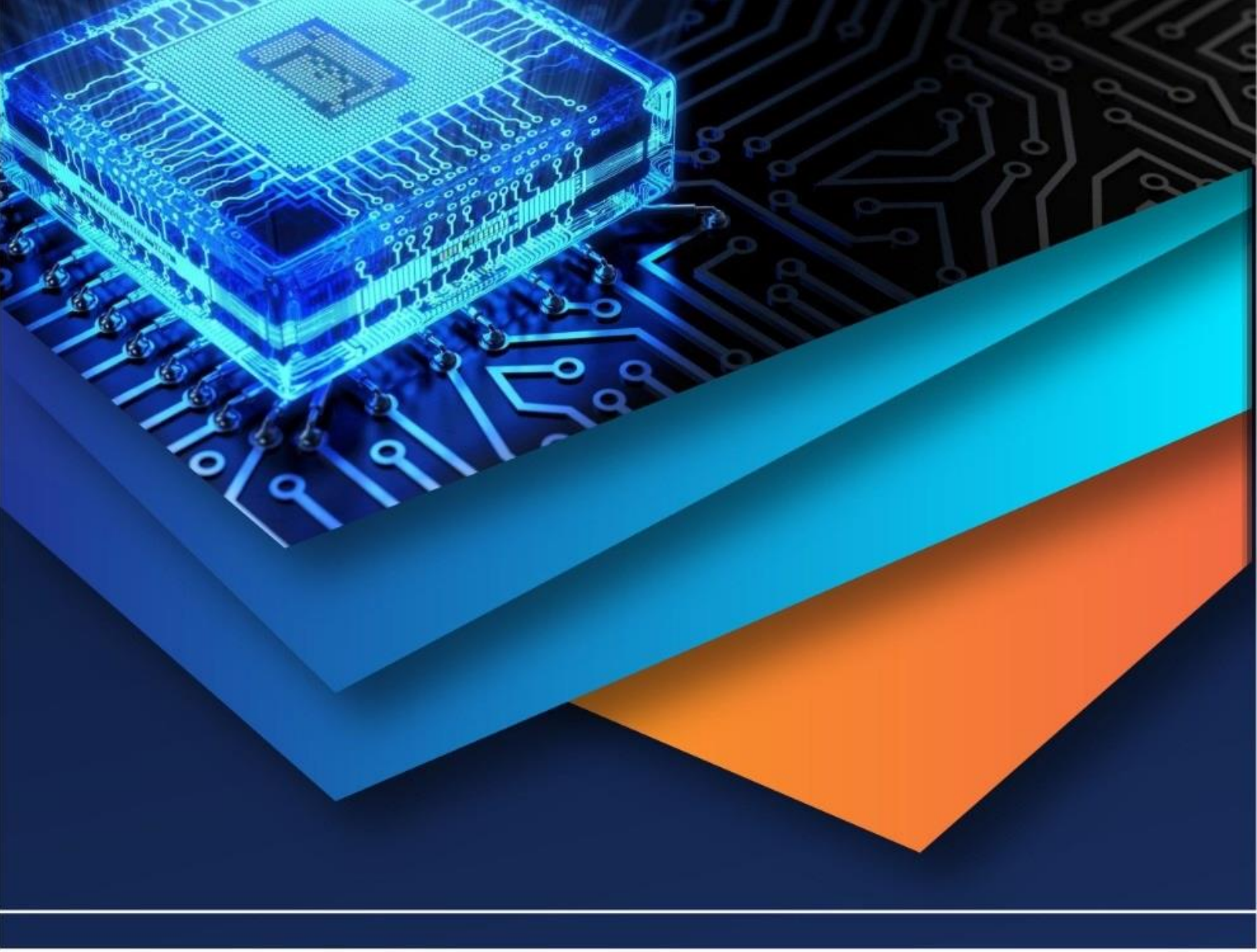
The authors would like to thank Mrs. P.N.L. Priyanka (M.Tech) for her valuable guidance throughout this work, and Dr. K. Chaitanya, Head of the Department of Cyber Security, Acharya Nagarjuna University, for providing the institutional support that made this research possible. The authors also thank OpenPhish and PhishTank for making their datasets publicly available.

REFERENCES

[1] A. K. Jain and B. B. Gupta, “A survey of phishing attacktechniques, defence strategies and research challenges,” EnterpriseInformation Systems, vol. 16, no. 4, pp. 527–565, 2022.
[2] M. Khonji, Y. Iraqi, and A. Jones, “Phishing detection: Aliterature survey,” IEEE Communications Surveys & Tutorials, vol.15, no. 4, pp. 2091–2121, 2013.
[3] A.Oest,Y.Safaei,A. Doupé,G.Ahn, B.Wardman,andK.Tyers,“PhishTime: Continuous longitudinal measurement of theeffectiveness of anti-phishing blacklists,” in Proc. USENIX SecuritySymposium, 2020, pp. 379–396.
[4] S. Garera, N. Provos, M. Chew, and A. D. Rubin, “A frameworkfor detection and measurement of phishing attacks,” in Proc. ACMWorkshop on Recurring Malcode (WORM), 2007, pp. 1–8.
[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for languageunderstanding,” in Proc. NAACL-HLT, 2019, pp. 4171–4186.
[6] Y. Lin et al., “Phishpedia: A hybrid deep learning based approachto visually detect phishing websites,” in Proc. USENIX SecuritySymposium, 2021, pp. 3793–3810.
[7] T. Fette, N. Sadeh, and A. Tomasic, “Learning to detect phishingemails,” in Proc. World Wide Web (WWW), 2007, pp. 649–656.
[8] L. Liu et al., “Cloudy with a chance of breach: Forecasting cyberthreats,” in Proc. USENIX Security Symposium, 2015, pp. 455–469.

[9] Y.Li, Z.Yang,X. Chen,H. Yuan,and W. Liu,"A stackingmodelusing URL and HTML features for phishing webpage detection,"Future Generation Computer Systems, pp. 27–39, 2019.
[10] P.Veličković,G.Cucurull,A.Casanova,A.Romero,P.Liò,and Y. Bengio, "Graph attention networks," in Proc. InternationalConference on Learning Representations (ICLR), 2018.





10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)