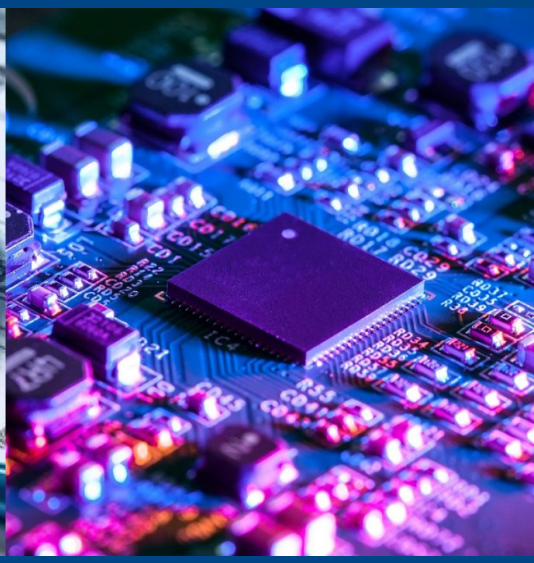
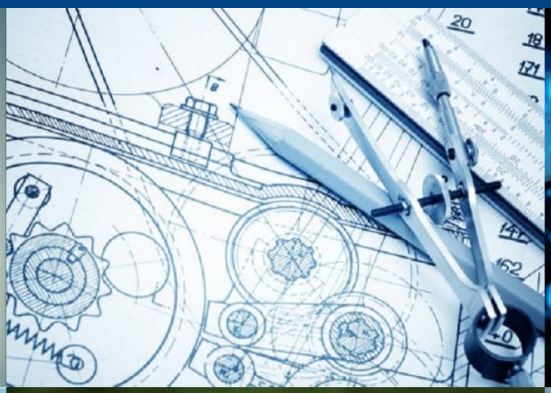
ISSN: 2582-7219

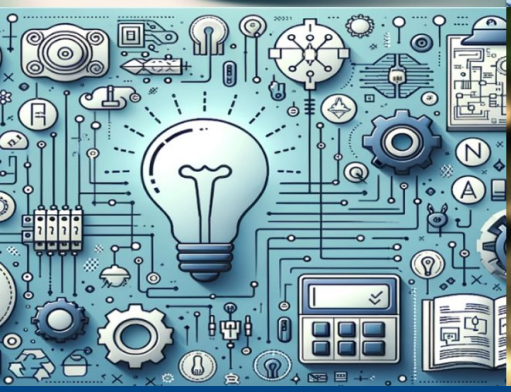




International Journal of Multidisciplinary Research in Science, Engineering and Technology

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



Impact Factor: 8.206 **Volume 9, Issue 4, April 2026**

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904354

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

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

An Explainable AI–Driven Multi-Model Phishing and Web Threat Intelligence

Periya Perumal P, Dr. G. Gomathi

Master of Computer Applications, Department of Computer Applications, B. S. Abdur Rahaman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India

Assistant Professor, Department of Computer Applications, B. S. Abdur Rahaman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India

ABSTRACT: Phishing campaigns have grown considerably more sophisticated making detection increasingly difficult for security system that rely ona single signal or method. A particular challenge is that malicious pages now routinely reproduce the visual and structural qualities of legitimate sites well enough to deceive both user and automated tools. This paper describes an AI driven detection framework that draws on both url level analysis and webpage screenshot inspection to identify phishing attempts. On the url side a machine learning classifier works from lexical and host based attributes extracted directly from submitted address. On the image side a CNN process screenshots to identify visual mimicry and deceptive layout patterns. Transparency is addressed through two complementary methods SHAP traces how individual url features shaped a given prediction while Grad-CAM produces a spatial map of the screenshot regions that most influenced the CNN output. A KRR module then applies rule based inference over both sets of predictions to improve reliability and reduce false positives. Evaluation results indicate gains in detection accuracy alongside meaningful improvements in interpretability which matters for practical adoption in security operation contexts.

KEYWORDS: phishing detection, explainable AI, convolutional neural network, machine learning, SHAP, Grad-CAM, web threat intelligence, KRR.

I. INTRODUCTION

Phishing one of the most frequently reported cybercrimes and remains a serious concern for internet users around the global. Cyber attackers commonly target individuals company and government organizations by developing fake websites or sending misleading emails that appears very similar to legitimate services. Because these websites often look trustworthy user may unknowingly share sensitive information such as username, passwords and financial details or other personal data. Several cybersecurity reports indicate that phishing is still a common starting point for major security incidents including data breaches and ransomware attacks conventional protection methods including blacklist based filtering and signature matching are often unable to recognise newly created phishing websites especially zeros day attacks that have not been previously recorded [5],[20]. For this reason researchers have increasingly focused on machine learning approaches for detecting phishing attempts. Techniques such as logistic regression and support vector machine examine url information by analysing its textual and structural properties to determine whether a website is legitimate or malicious [13],[22]. However approaches may face difficulties when attackers design webpages that visually resemble real websites which makes detection more complex.

Researchers have recently explored the use of deep learning techniques to analyse webpage screenshots for identifying phishing websites. Convolutional Neural network are commonly used because they can recognize visual similarities such as logo color themes and layout structures thatclosely match legitimate websites [11]. Many phishing pages are designed to look almost identical to trusted platforms so that userss may not easily suspect any risk and may provide confidential information . While url based approaches focus on text related properties image based methods provide additional support by capturing visual clues that cannot be observed from the url alone. Even though deep learning models often achieve good performance their internal working processes is not always easy to interpret. In cybersecurity understanding why a model gives a particular prediction is important for reliability and practical usage. Explainable AI methods such as SHAP and Grad-CAM are therfore used improve transparency. SHAP helps analyse

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904354

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

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

how different input feature influnce the predictions results where as Grad-CAM visuallly indicates the important areas of webpage image that contribute to CNN base classification decision [9],[10].

II. LITERATURE REVIEW

Research has extensively used the conventional machine learning and deep learning approaches for the detection of phishing attacks specifically for combating more sophisticated attacks on the web. Earlier approaches include the use of blacklisting and heuristic approaches that only detect well known malicious websites. However the approaches were not effective in detecting zero day attacks [5],[20]. For the purpose of overcoming the limitations of the earlier research used unsupervised learning approaches such as support vector machine and random forest for the detection of phishing attacks [13],[22]. These approach analyze the lexical and host based features of the urls. several research have used the approaches of extracting features from the url specifically the characteristics of the url such as the length of the url suspicious tokens and domain character [4],[18]. Several studies have also proved that the analysis of the features improves the detection of phishing attacks [14],[16]. However the attackers are modify the urls in such a way that they are similar to the legitimate websites.

visual analysis of webpage screenshots has become one of the more interesting direction in phishing detection especially given how convincingly modern phishing sites replicate legitimate platforms. CNN are well suited to this problem they can pick up on layout patterns login form structure and brand signal like logo or color choices that a user would instinctively recognize [11]. URL based features alone tend to miss these visual which is presumably why screenshot based approaches have attracted growing research interest. There has also been some work on hybrid architectures recurrent convolutional network being one example though whether these consistently outperform simpler CNN baselines is not always clear from the literature The bigger unresolved issue is that most of these models are essentially black boxes. In a security operations context that is areal problem an analyst who cannot see why something was flagged is unlikely to act on it confidently and false positives without explanation trust quickly. SHAP has been applied to address that at the feature level offering some visibility into how different inputs are weighted [10]. Grad-CAM servers a similar function for image based models drawing attention to the visual regions that drove a particular classification[9]. A KRR offer a somewhat different angle introducing rule based logical structure alongside learned representation. Each of these address part of the problem but they rarely appear in combination. Most published work still treats detection accuracy as the primary goal with interpretability handed separately if at all. Closing that gap building something that actually integrates detection explanation and reasoning into a coherent system remains somewhat suprisingly an open problem.

III. PROPOSED SYSTEM

The System being proposed here brings together several detection components under one frame work combining url based machine learning with image based deep laerning in the hope that the two approaches will compensate for each other blind spots. The underlying assumption is that neither url features or visual analysis alone is sufficient against phishing attacks that continue to using both together should in principle make the system harder to fool. URL processing happens first and features are pulled from the raw url itself things like length and suspicious keyword patterns and structural anomalies that tend to correlate with malicious intent. A logistic regression classifier than process these to produce an initial verdict. SHAP is applied at this stage specifically to keep the decision readable the goal is that someone looking at flagged url can actually see which features pushed the classification in that direction rather than receiving a probability score with no further explanation. The second component works on a screenshot of the webpage rather than the url. A CNN process the image to detect visual copied logos and layout structures that closely resemble known legitimate sites design choices that are meant to deceive at first glance. Grad-CAM runs alongside this to produce a visual map of which parts of the screenshots actually drove the model decision which helps with both interpretation and debugging.

A KRR module sits across both components applying rule based logic to cross check what models has flagged. It is about improving raw accuracy and more about adding a layer of structured reasoning that can catch cases where statistical models are uncertain or where specific known patterns should trigger a firm decision. The final output draws on all three components together rather than relying on any single model.

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904354

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

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

A. PROBLEM DEFINITION

Phishing attacks have grown considerably more sophisticated over the past few years and tools traditionally used to catch them are struggling to keep up. Blacklist filtering works fine for known malicious domains but a freshly registered phishing site will sail straight through it. Signature based detection has the same fundamental problem it recognize what it has already seen which is not particularly useful when attackers are constantly registering new domains or making small structural changes specifically to avoid triggering existing rules. Machine learning approaches that focus on a single signal have their own vulnerabilities. A model trained purely on url features can be beaten by an attackers who keeps their url clean and puts all deception into visual design of the page. Flip that around a model that only analyses page content can be tricked when attackers swap out text or randomise copy to break patterns matching. Neither approach is robust on its own and in practice attackers seem well aware of these limitations.what tends to get less attention in discussions about detection accuracy is how unusable many of these systems are for the people actually running them. A security analyst who receives a phishing confidence 94% result with nothing else to go on has to either trust the system blindly or investigate from scratch. Over time unexplained false positives particularly tend to confidence in the tool altogether and analyst start overriding or ignoring alerts. That is arguably as serious a failure mode as missing attacks outright. What seems genuinely missing from most current approaches is a system that treats detection and explanation as equally important problems handles multiple feature types simultaneously and designed from the start to remain useful as attack shift rather than becoming stale within months of deployment.

B. IMPLEMENTATION

The detection pipeline described here was designed around the idea that url signals and visual signals to fundamentally different kinds the evidence and that treating them separately rather forcing them into a single merged input gives each model the best chance of doing its job properly. Data collection on publicly available datasets containing both phishing and legitimate examples with url records and webpage screenshots handled as separate inputs from start. Preprocessing reflected this urls went through cleaning and normalization to strip noise and isolate structural components while screenshots were resized pixel normalized and filtered to apply bring inputs to a consistent format before hitting the model. Features extraction for the url side was done by hand computing lexical and host based indicators url length special character counts and domain age and ssl status and suspicious token patterns are converting these into numerical vectors. The image side takes a different approach entirely and relying the CNN to learn visual features directly from the data rather than specifying them in advance. This turns out to matter practically because the kinds of visual deception a CNN picks up logo mismatches structural mimicry misleading layout choices are difficult to define as explicit features but relatively easy to learn from examples.

Both modify were trained independently and evaluated against standard classification metrices accuracy and precision and recall and f1 score. Explain ability was built in at each stage rather than added afterwards. SHAP was applied to the logistic regression model to show which url features were driving individual predictions and Grad-CAM was used with cnn to highlight the image regions that most influence each classification decision.

The krr module sits at end of this pipeline taking the outputs and explanation results from both models and applying rule base inference to catch cases where statistical confidence is low or where specific known patterns override a borderline prediction. The intent was less improving accuracy numbers and more about reduce the false positive rate i ambiguous cases which tends to be where analyst trust breaks down fastest.

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904354

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

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

IV. SYSTEM ARCHITECTURE

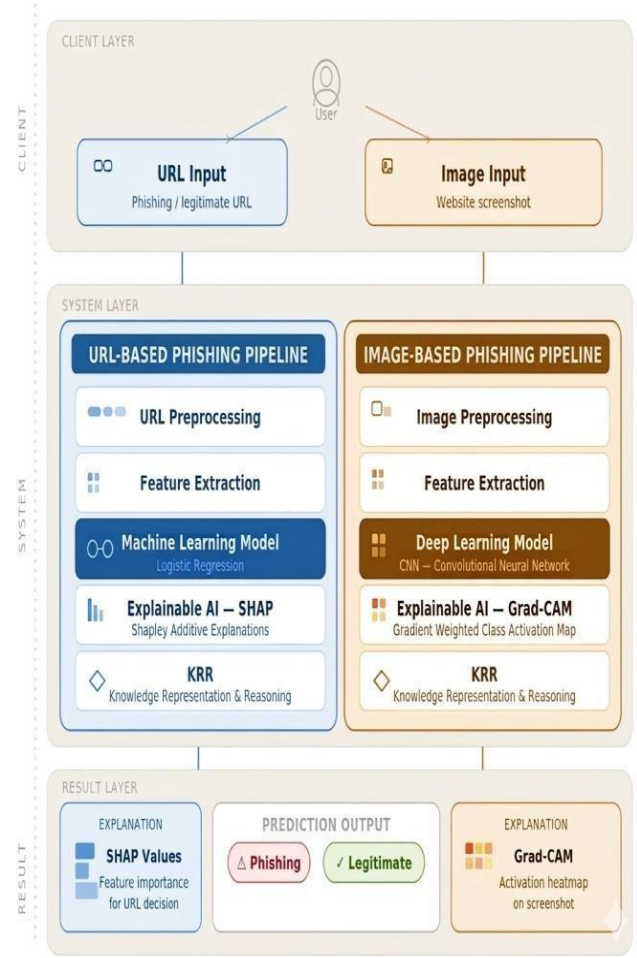


Figure 4.1: Architecture Diagram

The framework is built around two parallel pipelines that never actually merge, one working on URL and another screenshot of the page. Keeping them separate was deliberate choice. Combining URL features and image features into a single input vector is technically possible but it tends to obscure what each signal is actually contributing and make the systems harder to debug when something goes wrong. The URL site starts with whatever the user submits and run it through a cleaning step before anything else happens like lowercasing, protocol stripping, fragment removal, domain extraction. This sounds minor but matters because raw URL's are inconsistently formatted and small variations can throw off feature calculations. Once cleaned the system computes a fairly wide set of structural and host based attributes like URL length, dot count, presence of suspicious words whether an IP address appears in a place of a domain name, WHOIS registration age, SSL certificate status, redirect count, subdomain depth. The logistic regression classifier takes these as a numerical vector and returns a binary verdict then SHAP returns over that decision to show which feature actually moved the needle which is something that returns out to be genuinely useful when analysts want to understand why a particular URL was flagged rather than just that it was.

The image pipeline is structurally similar but operates on different scenario entirely like screenshots go through resizing, normalisation, contrast adjustment and noise filtering with OCR pulled into extract any text present in the image. This feature extraction here is looking for things like logo similarity to know legitimate brands, layout consistency, text alignment irregularities and invisible misspellings where these kinds of visual shortcuts that phishing pages tend to take when copying a legitimate site appearance. This CNN handles classification and Grad-CAM produce

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904354

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

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

a heat map overlay showing which regions of the screenshot of the model weighted most heavily. So generally in practice this tends to land on logos, form fields and header areas which aligns reasonably well with where visual deception usually happens.

Both the pipelines feed into a shared KRR module before anything is finalised. This rule based logic here acts as a sanity check rather than primary classifier it is mostly useful in cases like where one pipeline is confident and the other is not or where specific known patterns should trigger a firm decision regardless of what the probabilistic models say. The goal was a system where the reasoning behind the given prediction can actually be followed by a human analyst not just reported as a confidence score.

V. TECHNIQUES USED

C. SHAP

SHAP(Shapley additive explanations) which comes from game theory originally specifically from Shapley values which were developed to fairly distribute payoffs among players in a co- operative game. This adaptation to machine learning is conceptually straightforward instead of players you will have features and instead of payoffs you will have contributions to the models prediction. Each feature gets assigned a value that reflects how much it pushed the output up or down relative to a baseline.So in this system SHAP is applied specifically to the logistic regression classifier in the URL pipeline once features are extracted from a submitted URL like length, dot count, subdomain depth, special character frequency, presence of terms like "login" or "verify" the classifier will produce a prediction but that prediction on its own tells an analyst very little so the probability score of 0.87 phishing is not particularly actionable without knowing what drove it there.What SHAP adds is a breakdown of the score by feature where URL that is unusually long, contains multiple subdomains and includes words commonly associated with credential harvesting will show high positive SHAP values on those features which means they pushed the classification toward phishing while a clean domain name or valid SSL certificate might pull in the opposite direction. The summary plots and force plots make this readable at a glance rather than requiring someone to interpret raw numbers.The practical value of this is less about the model and more about the analyst. usually phishing detection systems tend to lose user trust when they flag things without explanation particularly when the flagged item looks superficially legitimate, where seeing which specific features triggered a prediction makes it possible to cross check the models reasoning against known phishing characteristics and importantly to catch cases where the model is technically correct but for the wrong reasons.

D. Grad-CAM

Grad-CAM(Gradient weighted class activation mapping) which works by tracing gradients back through a CNN to its final convolutional layer and using those gradients to figure out which spatial regions of the input image the network was actually paying attention to when it made its decision. The output is a heatmap overlaid on the original screenshot with warmer colours marking areas that had the strongest influence on the classification.The motivation for using it here comes from a fairly obvious limitation of CNNs they can be remarkably accurate while being completely opaque about why. The model that correctly identifies a phishing page but cannot communicate what it found suspicious is harder to trust and harder to improve. If a misclassification happens then there is no obvious place to start looking.In the image pipeline the CNN processes a webpage screenshot and classifies it as phishing or legitimate based on learned visual patterns with things like how a login form is positioned, whether a logo matches what it is supposed to look like or whether the overall layout resembles a known legitimate site closely enough to be suspicious. Grad-CAM runs after the classification and produces a heatmap showing where in the image the model concentrated it's attention. For example, the fake PayPal login page might produce a heatmap that lights up around the logo and the credential input fields which is exactly where a human analyst would look first.Whether this always maps neatly onto genuine reasoning is worth being honest about. Heatmaps show where gradients were large not necessarily where the model understood something meaningful. But in practice for visual phishing detection the highlighted regions tend to be interpretable and roughly consistent with what a human would flag which makes the tool useful even if it is not a perfect window into the models internal logic.

E. Knowledge Representation and Reasoning (KRR)

KRR sits slightly outside the mainstream of modern machine learning which tends to be sceptical of hand coded rules in favour of learned representations. The argument for including it here is practical rather than theoretical where there are things that domain experts know about phishing that do not need to be learned from data and encoding that knowledge directly is faster and more reliable than hoping a statistical model picks it up.

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904354

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

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

The module works as a post prediction layer so by the time it runs both the URL classifier and the CNN have already returned their verdicts. The KRR does not replace those verdicts it will cross verifies them against a set of rules derived from established phishing characteristics and security guidelines. The URL that scored borderline on the logistic regression but also happens to have a domain registered within the last week, no SSL certificate and a suspicious keyword in the path is going to look very different under rule based scrutiny than a borderline URL that fails none of those checks. These rules are essentially operationalise and this kind of reasoning an experienced analyst would apply manually.The image feature works similarly if a Grad-CAM has flagged the login section of a page as region driving the phishing classification and the rules contain patterns associated with credential gathering pages then those two signals together carry more weight than either of alone and the confidence level adjusts accordingly.In general rule based system have known weakness where they are only as good as the rules they contain and attackers who understand the ruleset can design around it but for the specific purpose of reducing false positives in ambiguous cases rather than serving as a primary detection mechanism a well maintained KRR module is genuinely useful so the goal here was never to replace the learned models but to add a layer of structured reasoning that keeps the system output grounded in what the security community actually knows about how phishing works.

VI. EXPERIMENTAL EVALUATION

Testing the system means evaluating two pipelines that operate independently and serve somewhat different purposes so treating them as a single combined model for evaluation purposes would be misleading. The URL pipeline running submissions through logistic regression, SHAP interpretation and the KRR layer which was assessed separately from the image pipeline which follows the CNN and Grad-CAM stages through the same KRR module before producing a final verdict.

Standard classification metrics were used throughout for accuracy, precision, recall and F1- score. These are familiar enough in the literature that they need little justification though it is worth noting that in a phishing detection context recall tends to matter more than precision in most deployment scenarios, so missing a phishing site is generally a worse outcome than a false positive even if both are undesirable. Beyond the headline metrics the evaluation also looked specifically at what the KRR module was doing to the predictions it received since one of the system core claims is that rule based reasoning should improve reliability rather than just add complexity.

Metric	Value (%)
Accuracy	95
Precision	94
Recall	92
F1-Score	93

TABLE.1 URL Performance

The URL pipeline results are summarised in Table
1. The logistic regression reached 95% accuracy on the test set which is reasonable for a model working purely from lexical and structural features where there is no page content, no visual analysis, just only what can be extracted from the URL string itself. Precision came in at 94% which means the false positive rate was relatively contained though not negligible. Recall was slightly lower at 92% which in practice means roughly one in twelve phishing URLs was not caught at this stage which is something worth flagging given that the URL pipeline is the first line of detection rather than the last.

The F1-score of 93% sits where you would expect given those two values What these numbers do not capture is how the model behaves on edge cases like URL's that are structurally clean but semantically suspicious or legitimate URL's that happen to share features with known phishing patterns, so those limitations are part of why the image pipeline and KRR layer exist.

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904354

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

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

Metric	Value(%)
Accuracy	93
Precision	91
Recall	90
F1-Score	90

TABLE.2 Image Performance

Table 2 presents the performance of the image pipeline where the CNN achieved 93% accuracy which is few points behind the URL classifier because of visual classification. The webpage layouts vary enormously across the legitimate site and phishing pages that closely copy a well known brand are genuinely difficult to distinguish from the real thing at the pixel level. The result displays precision was 91% and recall 90% which is giving an F1 of 90%.

The recall result performance should be looked closely. A 10% miss rate means one in every ten visually deceptive pages getting through is a meaningful gap and it points to something which metrics alone cannot explain where CNN Is likely struggling most on high effort phishing pages. The ones where someone has put real work into replicating a target site appearance rather than throwing together something generic then those cases are also the most dangerous ones, so whether the KRR layer recovers some of those misses is part of what the combined evaluation is meant to show.

Pipeline	Without KRR Accurac y	With KRR Accura cy	Improvemen t
Url-LR	92%	95%	+3%
Image- CNN	90%	93%	+3%

TABLE.3 KRR Impact Performance

Table 3 looks at what the KRR module actually adds once dropped into the each pipeline. The URL classifier displays 92% accuracy before the reasoning layer and jumped to 95% afterwards so a 3% point gain that sounds modest but represents a meaningful reductions in misclassified cases at the scale. The image pipeline showed the same 3% improvement where it jumped from 90% to 93%

what these numbers suggest is that the rule based layer is doing genuine work rather than just adding extra things. The cases catches are the ones where the statistical model was uncertain whether borderline predictions a URL or image which sits close to the decision boundary and known phishing patterns encoded in the rules are enough to tip the verdict in the right direction which holds across attack types not well represented in the training data is harder to say from these results alone and would need more targeted testing to establish properly.

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904354

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

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

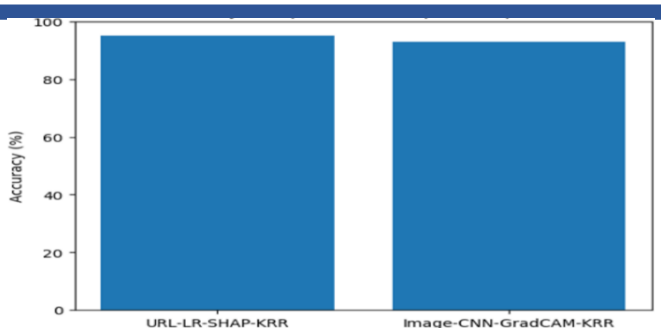


Figure 6.1: Url And Image Accuracy

The bar chart puts the two pipelines side by side and shows 95% for the URL classifier and 93% for the image based classifier. The gap is small enough that it probably should not be over interpreted the 2% points across different input types and different model architectures does not tell much about which approach is fundamentally better like what it does suggest is that structured featured extraction from URl strings which is slightly more reliable under these conditions which makes intuitive sense given how much variation exists in webpage visual design compared to the relatively constrained space of URL patterns.

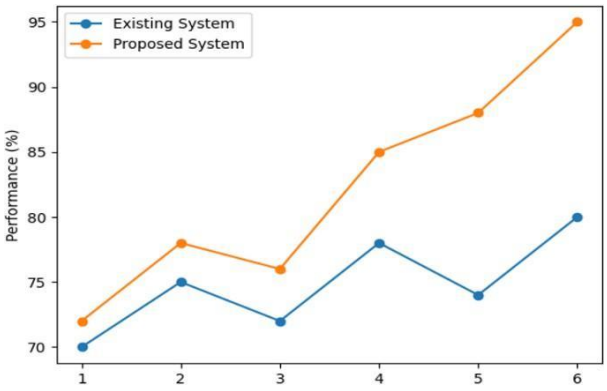


Figure 6.2: Existing System And Proposed System

The line chart tracks both systems across evaluation stages and the gap widens consistently rather than appearing at a single point. The existing system stays between 70 and 80% band throughout. The proposed system starts at 72% but climbs to 95% by the final stage this trajectory matters more than the endpoint alone. The system that improves steadily as more components are added is behaving as designed whereas a large jump at one specific stage would raise questions about where the gain is actually coming from.

The results hold up reasonably well against what the framework was supposed to do. Detection accuracy improved the explainability components produced output that are at least interpretable by a human analyst and the KRR layer contributed a measurable lift rather than sitting inert at the end of the pipeline Whether these numbers translate cleanly to real-world deployment is a different question and controlled evaluation conditions tend to flatter the every system and phishing campaigns in practice are specifically designed to avoid whatever the current detection methods are looking for. The combined architecture made what a single model baseline cannot and it gives analysts something to work with beyond a bare classification score.

VII. RESULTS & DISCUSSION

By running these two pipelines separately will be a good and perfect design because it let's each model run on it's own trained model like where the URL side pipeline handles the structured lexical patterns while the image side pipeline focuses on layout analysis. By keeping them apart we can avoid the bad output which often happens when you forcefully run two different data types to merge. The URL pipeline is more accurate but that makes sense the URL

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

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

structures are generally more consistent than webpage designs which can be look like anything these days.

The image pipeline really shines where the URL model fails specifically on pages where the link looks boring but the visual design is a total brand rip off. By using Grad-CAM here is not just about filling the accuracy statistics but about transparency. If an analyst can see exactly which logo or text box triggered the alert then they can make a much faster action on whether to take the site down. As for the KRR module the results were clear. A 3 point accuracy boost across the board just by applying logical rules shows that we are actually resolving ambiguity not by just getting lucky. These rules will need to be tweaked as phishing tactics evolve which is an extra layer of work that doesn't show up on a performance chart.

F. ADVANTAGES

A few things distinguish this setup from more conventional detection approaches some more significant than others. The separation of URL and image analysis into independent pipelines means either of the component can be deployed on its own so an organisation without the infrastructure to process screenshots at scale can still run the URL classifier and the results suggest that value alone is effective. This kind of modularity tends to get designed out of system that prioritise raw performance over practical deployment so it seems to be worth preserving here.

The transparency argument is harder to quantify but probably matters more in practice. Detection accuracy figures look good in evaluation papers and mean relatively little to a security team drowning in alerts. The SHAP outputs and Grad-CAM heat maps do not solve the alert fatigue problem but they will give analysts something to anchor a decision to rather than forcing a binary trust or just ignore response to every flag the system raises.

The zero day case is worth being careful about since, the system analyses structural and visual patterns rather than matching against known malicious signatures which does give it some ability to catch attacks it has never seen before but some ability is doing a lot of work in that sentence where a sufficiently novel phishing campaign that does not resemble anything in the training data will still cause problems and the KRR rules will not help if the attack was designed with those rules in mind. These are the genuine limitations rather than just theoretical ones.

G. FUTURE ENHANCEMENTS

There are several ways that we could look from here even though they vary in how straight forward they would have be to be implement. The most immediate win would be integrating real time threat intelligence things like live blacklist lookups and domain reputation scores. Right now the current system makes decisions based on what it was trained on and what the rules encode just by adding a live data stream would let it respond to domains that have been flagged in the past few hours rather than the past few months.

URL pipeline feature can also be extend bit further from the current one. We could look into entropy based measures and behavioural traffic signals capture things that lexical features miss, so the URL can look structurally normal while the traffic pattern around it looks nothing like legitimate use Whether that complexity is worth the added data collection requirements depends on the deployment context but for enterprise environments it seems like a reasonable trade.

Transfer learning is the obvious next step for the image side. The CNN here was trained from a relatively constrained dataset where a model pre trained on a much larger corpus of webpage screenshots would likely handle unusual or complex layouts better particularly for phishing pages targeting less common platforms or non- English speaking users. Multilingual phishing is genuinely under represented in most detection research and probably deserves more attention than it gets.KRR module has biggest practical weakness which is that the rules are static where someone has to update them manually as attack patterns shift which creates a lag and an ongoing maintenance burden. An automated rule learning component that derives new rules from misclassified cases would address this though building something reliable enough to trust in a security critical context is non-trivial that feels like the most interesting open problem the system leaves behind.

VIII. CONCLUSION

When we started building this system the core idea was simple like most phishing detectors fail because they only look at one thing. If you only check the URL then well designed fake page slips through. If we scan only the visual layout then clean-looking link fools the model so we decided to combine both features and add explainability on top to use

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

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

rule based reasoning as a final check.Based on the results it works as expected. Detection accuracy went up across every metric compared to single model approaches. The KRR module has been improved consistently on both pipelines rather than just sitting there doing nothing. The SHAP and Grad CAM output were actually readable by a human analyst which honestly is not always the case with explainability tools where some of them technically work but produce output that nobody knows what to do with.Now, we're not going to pretend the system is perfect. These tests were run under controlled conditions and real phishing attacks are specifically designed to prevent whatever detection tools are currently out there. We are not fully know how this holds up when attackers are actively trying to break it.But here is what we are confident about by using this approach it covers more ground than a single signal system and it gives analysts the actual context instead of just a percentage score and each pipeline still works independently as per the design so this is a practical advantage not just a theoretical one Whether it properly delivers in a live security environment is the next thing we have to find out.

REFERENCES

[1] A. K. Jain and B. B. Gupta, "Phishing detection: Analysis of visual similarity based approaches," Security and Communication Networks, vol. 10, no. 8, pp. 1448–
1464, 2017.
[2] M. Marchal, J. Francois, R. State, and T. Engel, "PhishStorm: Detecting phishing with streaming analytics," IEEE Transactions on Network and Service Management, vol. 11, no. 4, pp. 458–471, 2014.
[3] S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A framework for detection and measurement of phishing attacks," ACM Workshop on Recurring Malcode, 2007.
[4] R. Verma and A. Das, "What's in a URL: Fast feature extraction and malicious URL detection," IEEE CNS, 2017.
[5] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites," ACM SIGKDD, 2009. U.
A. A. Arachchilage and S. Love, A ng threat avoidance perspective," Computers in Human Behavior, 2014.
[6] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.
[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks," ICLR, 2015.
[8] R. R. Selvaraju et al., "Grad-CAM: Visual explanations
[9] from deep networks," IEEE ICCV, 2017.
[10] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," NeurIPS, 2017.
[11] A. Rajab et al., "Phishing website detection using deep learning," Future Generation Computer Systems, 2020.
[12] M. Aburrous, M. A. Hossain, F. Thabatah, and
[13] K. Dahal, "Intelligent phishing detection system," Expert Systems and Applications, 2010.
[14] B. B. Gupta et al., "Machine learning-based phishing detection," Journal of Information Security and Applications, 2018.
[15] D. Sahoo et al., "Malicious URL detection using machine learning," IEEE BigData, 2017.
[16] T. Fawcett, "An introduction to ROC analysis," Pattern Recognition Letters, 2006.
[17] N. Sahingoz et al., "Machine learning based phishing detection," Applied Soft Computing, 2019.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com