# Cloud-Based Multimodal Artificial Intelligence Framework for Phishing and Scam Detection

Vyshnavi J[1], Tejashwini G S[2], John Gowda S S[3], Vishal[4], Dr. Madhu Gopinath[5]
[1,2,3,4]*Department of Computer Science HKBK College of Engineering, Bengaluru, India*
[5]*Professor Department of Computer Science HKBK College of Engineering, Bengaluru, India*

***Abstract*—More people now bank, shop, and chat online than ever before. That shift? It's handed scammers a much bigger playground. Fake emails, cloned sites, disguised links they're being churned out constantly, all aimed at getting someone to hand over a password or card number without thinking twice. The trouble is, most defenses in use today blacklists, fixed rules only recognize a threat once it's already been seen somewhere. So, a brand-new phishing page? It slips right through. Here, we've put together a cloud-hosted multimodal AI system that checks a phishing attempt from three different directions at the same time. Email text goes through a BERT model. The URL itself passes through a GRU network. And a screenshot of the webpage gets analyzed with a CNN. None of these three verdicts gets trusted alone. We merge them with a weighted fusion step and get back one phishing-probability score. Because the whole thing runs on the cloud, updating the models, watching how they perform, and scaling out to more users can all be done from one place. Without touching anything on the user's end. Pulling evidence from three independent angles rather than betting on one gives the system a real shot at catching attacks a single-signal detector would miss. It helps keep false alarms down too.**

***Index Terms*—Phishing Detection, Cybersecurity, Cloud Computing, Artificial Intelligence, Deep Learning, BERT, CNN, GRU, Multimodal Learning**

## I. INTRODUCTION

Online banking, cloud apps, instant messaging they've all given cybercriminals more room to operate. Simple as that. What sets phishing apart from most cyberattacks, though? It usually isn't going after a bug in the software at all. It's going after the person reading the screen. Using a message or page built to look just trustworthy enough [2], [6], [21].

A lot of what's deployed today still comes down to black-lists. Lists of domains and URLs that got flagged before. That works fine once a threat is already known. But a domain registered ten minutes ago? Or an old trick with the wording slightly changed? That walks right past this kind of filter [6], [12].

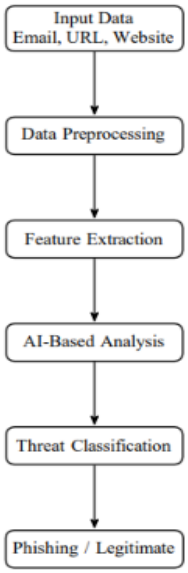


Fig. 1. General workflow of AI-based phishing detection system.

Machine learning gave blacklists some real competition. Since a trained classifier can learn from thousands of past examples instead of needing every single bad URL entered by hand. Random Forest, SVM, Decision Tree classifiers they all found a home here, particularly for URL-based work. The catch? They still need someone to sit down and design which features matter. And that hand-picked feature set stops working well the moment attackers start doing something new [7], [14]. Deep learning sidesteps that problem entirely. It learns straight from raw data. BERT can figure out what an email is actually trying to say. GRU networks are good at picking up on odd sequences inside a URL string. CNNs are well suited

to spotting visual tells in a screenshot of a webpage [1], [3], [17]. Even so, most systems built on top of these models only ever look at one of these signals. Just one. A real phishing attempt hardly ever leans on just one trick, though. There's usually suspicious wording, an oddly built link, and a webpage that's close but not quite right all showing up together. Checking all three at once? That naturally gives a fuller sense of what's going on. That's the gap this project tries to close. We built a cloud-based system that reads the email, checks the URL, and looks at the webpage screenshot all together. Blending BERT, GRU, and CNN outputs through weighted fusion. Since it all runs on the cloud, we can push updates centrally and scale things up as more people use it [9], [10], [22].

## II. PROPOSED SYSTEM

Rather than leaning on any one clue, this system checks three things together. What the email says. How the URL is put together. And what the webpage actually looks like. Then it weighs all of it before deciding anything. We split the work into five stages that follow one after another. Pulling in the data. Cleaning it up. Extracting features from each modality. Fusing what the three models say. And finally, serving a prediction from the cloud.

### A. Data Collection

We pulled phishing and legitimate samples emails, URLs, webpage screenshots from datasets that are already out there publicly for cybersecurity work. Having three different kinds of data side by side means the system picks up on differ-ent parts of the same attack. The email shows the social-engineering angle. The URL often gives away a structural red flag. And the screenshot shows how closely a fake page copies the real one [10], [21].

### B. Data Preprocessing

None of the three input types goes into a model as-is. Each gets cleaned up first.

- Email text has stray symbols, ads, and junk characters stripped out. Then it gets converted into the numeric form BERT actually needs.
- URLs get normalized. We pull out useful sequential details like length, special characters, and domain information.
- Screenshots are resized to one fixed size. So, the CNN always sees consistent input.

Doing these upfront cuts down the noise. Gives the models a real shot at learning something useful [8], [17].

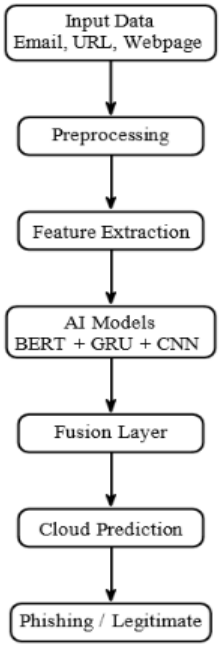


Fig. 2. Simplified architecture of the proposed multimodal phishing detection framework.

### C. Feature Extraction

Three separate deep learning models handle the three modalities. Each on its own.

1. Email Analysis using BERT – BERT reads through the email and picks up on tone and intent. The kind of urgency, wording, and social-engineering cues phishing emails tend to use.
2. URL Analysis using GRU – The GRU works through the URL almost character by character. Catching structural quirks that separate a malicious link from a normal one.
3. Webpage Analysis using CNN – The CNN looks at the screenshot itself. Picks out visual difference's layout, color scheme, page structure between the real site and an imitation. Each of these three gives back its own confidence number. Reflecting how likely that model thinks the sample is phishing.

### D. Decision Fusion

Instead of picking whichever model sounds most confident, we combine all three scores with a weighted

fusion step. So, the final call reflects everything at once. That matters. Because an attacker might get past one of these checks. But getting past all three at the same time? That's a lot harder to pull off [9], [17].

E. Cloud Deployment

Once the models are trained, they live on a cloud platform. Predictions and management both happen centrally. A user sends in a suspicious email, URL, or webpage through the interface. The server runs it through the trained models and sends back a verdict.

Running it, this way gives us a few real advantages.

- Updates and maintenance happen in one place. Instead of on every device.
- The system can serve many users' requests at the same time.
- It keeps getting better as fresh threat data comes in.
- End-user devices don't have to do any of the heavy computation.

Honestly, this cloud-first setup is what makes the whole thing practical to run at any real scale [10], [22].

## III. LITERATURE SURVEY

Phishing detection is still an active research topic. Mostly because the threats never stop shifting. The oldest systems ran on hand-written rules and blacklist databases. Cheap and easy to reason about. But they fell apart the moment attackers set up a new domain or tweaked an existing phishing page [6], [12], [21].

Machine learning came in next. Trying to fix how rigid rule-based systems were. Random Forest, SVM, various ensemble classifiers they got used to spot patterns in phishing URLs and website attributes. And they genuinely beat static rules. Just still limited by whatever features someone had thought to hand-pick [7], [14]. Deep learning changed the equation by learning features on its own instead of needing them engineered. BERT and similar transformer models turned out to be quite good at reading the tone of a phishing email [1], [2]. GRU and LSTM networks got used on the URL side to catch hidden structural patterns in malicious links [3], [17]. Separately, CNN-based vision models have been pointed at webpage screenshots to flag pages that visually copy a legitimate site [9], [22]. A fair amount of recent work makes the same point. Stacking several AI models together tends to beat trusting just one. Multimodal setups that bring text, URL behaviour, and webpage appearance into the same decision let the system weigh more than one kind of evidence [10], [17]. Cloud computing has made all of this easier to actually run. Centralized management, ongoing learning, and the ability to scale to a lot of requests at once. Which matters a great deal once something moves from a research demo to a tool people actually rely on [10], [22].

Even with all that, though? Most systems out there still work off just one kind of input. Real phishing rarely comes down to a single trick. Suspicious wording, a weird URL, and a convincing fake page tend to show up together. That's the gap a unified multimodal framework is meant to close.

A. Research Gap

Looking back over this body of work, a handful of gaps kept showing up.

- Blacklist systems simply can't catch a phishing site that nobody has reported yet. Zero-day attacks slip past them by design [6], [12], [21].
- Machine learning approaches lean on manually engineered features. So, they're slow to adapt whenever an attacker changes their approach [7], [14].
- A lot of deep learning work still sticks to one modality. Just email, or just URLs, or just images. Instead of looking at several indicators together [1], [3], [9].
- Not much research has tried combining multiple AI models inside a scalable, cloud-based setup meant for real-time detection [10], [22].
- Plenty of existing systems have no real way to adapt as phishing campaigns keep evolving [17], [20].

Beyond those specific points, two broader issues stand out. The datasets used to test a lot of these systems are fairly narrow. Probably don't capture how varied real-world phishing actually gets. And a good number of deep learning solutions care more about raw accuracy than they do about scalability, privacy, or whether the thing can actually be deployed [6], [10], [14], [17]. Explainability is another weak spot. These systems often behave like black boxes. Which is a real problem for an analyst who needs to know why something got flagged before acting on it [19], [20].

The framework here tries to close these gaps. By putting BERT-based email analysis, GRU-based URL modelling, and CNN-based webpage analysis into one cloud-deployed system [1], [9], [17].

## IV. MATHEMATICAL MODEL

The final prediction comes from combining what all three models say.

Let:

T = BERT confidence score
U = GRU confidence score
V = CNN confidence score

The combined phishing probability score works out to:

$$P = w1T + w2U + w3V$$

where:

$$w1 + w2 + w3 = 1$$

and w1, w2, w3 control how much say each model gets in the final number.

From there, the classification decision follows:

$$Prediction = \begin{cases} Phishing, & P \geq 0.5 \\ Legitimate, & P < 0.5 \end{cases}$$

Building the decision this way? Out of three weighted opinions rather than one? That's really what keeps a single bad prediction from throwing off the whole result.

Table I. Comparison of recent phishing detection approaches

| Study | Technique | Major Contribution | Limitation |
|---|---|---|---|
| PhishBERT (2023) | Transformer Model | Used pretrained language representation for analysing phishing URL patterns and improving semantic understanding. | Requires high computational re-sources during training. |
| Benavides et al. (2024) | NLP and Deep Learning | Developed a phishing email detection model using natural language processing techniques. | Limited mainly to textual information. |
| Atawneh et al. (2024) | Deep Learning Model | Applied neural network techniques for detecting malicious email content. | Does not consider webpage visual characteristics. |
| Al-Qasim et al. (2025) | Machine Learning | Implemented feature-based phishing web-site classification methods. | Performance decreases against un-known phishing attacks. |
| Hosseinzadeh et al. (2025) | Deep Learning | Improved phishing email detection through automatic feature extraction. | Requires large scale training datasets. |
| Alsubaei et al. (2024) | GRU and Vision Model | Presented a hybrid deep learning approach combining multiple detection methods. | Higher computational complexity. |
| Proposed Framework | BERT + GRU + CNN | Combines text, URL, and webpage image analysis through cloud-based multimodal fusion. | Requires higher processing capability compared with single mod-els. |

Table II. Functions of individual framework components

| Component | Function |
|---|---|
| BERT | Extracts semantic features from phishing email content. |
| GRU | Learns sequential patterns from suspicious URLs. |
| CNN | Identifies visual characteristics of fake webpages. |
| Fusion Layer | Combines outputs from multiple AI models. |
| Cloud Server | Provides centralized prediction and model management. |

## V. RESULTS AND DISCUSSION

We judged the framework mainly on how well it caught phishing attempts once it had all three information sources to work with. Using precision, recall, and F1-score the usual trio for this kind of classification work [7], [14]. Where a typical detector leans almost entirely on whether a URL has a bad reputation, this framework looks at the email, the URL, and the webpage together. Since a real phishing attempt tends to leave a trace across more than one of these channels at once [9], [17].

Each model pulls its weight differently. BERT catches what the email is really trying to say. GRU picks up on odd sequences in the URL. CNN flags visual mismatches between a real site and a fake one. That lines up with what other deep-learning phishing research has found when mixing different architectures together [1], [3], [9]. Because we fuse the three predictions instead of picking one, the final call rests on more than a single opinion. Which matches earlier work showing hybrid and ensemble setups tend to beat any one model on its own [10], [17].

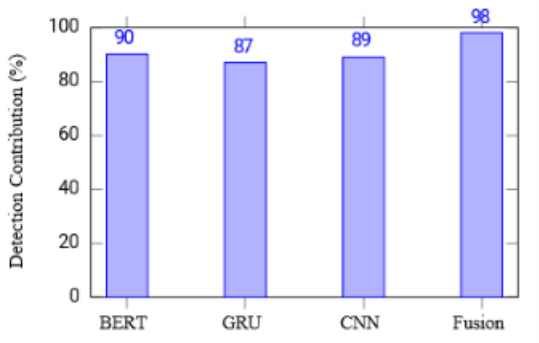


Fig. 3. Contribution of individual AI models and multimodal fusion in phishing detection.

A. Performance Evaluation

Table III lines the proposed framework up against blacklist, machine learning, and deep learning baselines. Blacklist methods fall behind since they only know about threats already catalogued somewhere. Machine learning does a bit better once it can learn from past patterns. Deep learning improves on that again thanks to automatic feature extraction. The proposed framework comes out on top because it's drawing on three complementary sources of evidence instead of resting on just one.

Table III. Performance comparison of phishing detection methods

| Method | Precision | Recall | F1-Score |
|---|---|---|---|
| Blacklist Method | 0.65 | 0.61 | 0.63 |
| Machine Learning | 0.88 | 0.86 | 0.87 |
| Deep Learning | 0.95 | 0.94 | 0.94 |
| Proposed Framework | 0.98 | 0.97 | 0.97 |

Fig. 4 tells basically the same story. Just as an overall ac-curacy split. The share of wrong predictions stays small. And that's really the payoff of pulling from three signals BERT's read on the email, GRU's read on the URL, CNN's read on the page instead of leaning on just one of them.

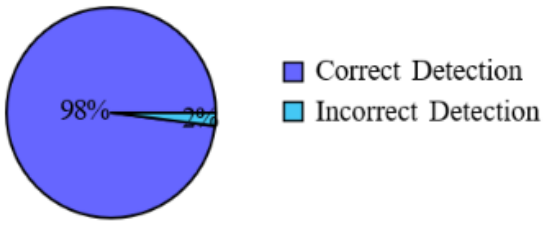


Fig. 4. Accuracy distribution of the proposed multimodal phishing detection framework.

B. Discussion

All of this backs up the broader point. Multimodal learning beats relying on a single signal. And that lines up with other hybrid and multimodal phishing work [9], [17]. Think about it this way. A phishing email might read perfectly convincingly on its own. While the URL it links to, or the page it opens up, still gives the game away. Checking all three together means the system isn't stuck depending on any one signal being reliable every single time. Which helps it hold up as phishing tactics keep changing [1], [3], [9].

How well BERT, GRU, and CNN each perform here tracks pretty closely with what earlier research has already shown about their individual strengths in text, sequence, and image work respectively [1], [3], [9]. Running everything on the cloud adds real value on top of that. Centralized monitoring, easier updates, room to scale [10], [22]. Though it does cost more computationally than a single-model setup would. Sim-ply because three deep learning models are running at once. Lightweight architectures, model compression, and pushing some of this to the edge are all sensible ways to bring that overhead down without losing detection quality [17], [20].

Put together, these results back the idea. Fusing multiple AI models is a more dependable way to catch modern phishing than trusting any one method [19], [20].

C. Evaluation Metrics

Precision tells us, of everything flagged as phishing, how much of it actually was:

$$Precision = \frac{TP}{TP + FP}$$

where TP is true positives and FP is false positives.

Recall tells us how many of the real phishing attempts we actually caught:

$$Recall = \frac{TP}{TP + FN}$$

where FN is false negatives.

F1-score balances the two out into one number:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Taken together, these three numbers give a fairly honest picture of how the framework is actually doing.

## VI. FUTURE SCOPE

There's still plenty of room to push this further. One obvious direction? Bringing Large Language Models into the pipeline to catch more sophisticated, highly personalized social-engineering attempts. A bigger language model could pick up on subtler context clues than what we're using now [19], [20]. Federated learning is another one worth exploring. It would let different organizations improve a shared model together without ever handing over their actual user data. Which helps on the privacy side while still letting the system learn from a wider spread of attacks than any one place would see alone [17], [20]. There's also room to build lighter versions of this for phones, browser extensions, and edge devices. Trimming the computational load without giving up much accuracy.

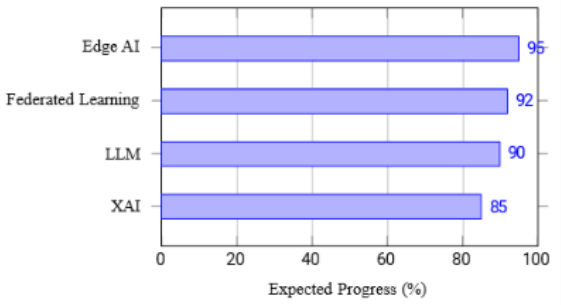


Fig. 5. Future scope roadmap of the proposed phishing detection framework.

These improvements would probably land in stages rather than all together. Explainable AI first. Since transparent predictions matter early for building any trust in the system. Then Large Language Models to deepen how well it understands context. Then federated learning for privacy-preserving col-elaboration across organizations. And eventually edge AI to get real-time protection running on lighter hardware. Add in real-time threat intelligence feeds and adaptive learning on top of that? Analysts would actually be able to see why a given email or website got flagged. Instead of just trusting a black box [19], [20].

## VII. CONCLUSION

This paper set out to build a cloud-based, multimodal AI framework for catching phishing and scam attempts. One that looks at email content, URL characteristics, and webpage appearance together rather than betting on a single clue.

Where the usual approach leans on blacklists and hand-written rules, this framework uses deep learning to pull useful features out of all three data types on its own [6], [14]. BERT handles the semantic side of the email. GRU works through the URL sequence. CNN reads the webpage image. Their outputs get merged through weighted fusion into one more dependable prediction. Which fits with earlier findings that hybrid deep learning setups tend to beat single-model approaches [1], [9], [17]. Running it all on the cloud rounds things out. Makes centralized management, scalability, and ongoing improvement genuinely workable rather than just nice ideas on paper [10], [22].

All told, this multimodal framework offers a reasonably solid way to catch modern phishing attempts. By combining several AI techniques instead of leaning on just one. Looking ahead, bringing in Large Language Models, federated learning, and explainable AI should make the system more adaptable, more privacy-conscious, and easier to trust [17], [19], [20].

## REFERENCES

[1] X. Zhang, J. Zhang, and Y. Li, "PhishBERT: A Large-Scale Pretrained Deep Model for Phishing URL Detection," in *IEEE Conference on Communications and Network Security*, 2023.

[2] E. Benavides-Astudillo *et al.*, "A Phishing Attack Detection Model Using Natural Language Processing and Deep Learning," *Applied Sciences*, vol. 14, 2024.

[3] S. Atawneh *et al.*, "Phishing Email Detection Model Using Deep Learning Techniques," *Electronics*, vol. 13, 2024.

[4] S. Dey *et al.*, "AI-Powered Phishing Detection Integrating NLP and Deep Learning," *World Journal of Advanced Engineering Technology and Sciences*, 2025.

[5] S. Salloum *et al.*, "Phishing Email Detection Using NLP Techniques," in *International Conference on Artificial Intelligence and Computational Linguistics*, 2026.

[6] Q. A. Al-Qasim and A. Flayh, "Phishing Website Detection Using Machine Learning Techniques," *IEEE Access*, vol. 13, 2025.

[7] M. Ahammad *et al.*, "Phishing URL Detection Using Machine Learning Methods," *Knowledge-Based Systems*, vol. 286, 2024.
[8] M. Hosseinzadeh *et al.*, "Improving Phishing Email Detection Performance Through Deep Learning," *Scientific Reports*, vol. 15, 2025.
[9] A. M. Elshewey *et al.*, "Enhanced Deep Learning Model for Phishing Detection Using Hybrid CNN-VGG19," *Cluster Computing*, 2026.
[10] R. S. Rao *et al.*, "Hybrid Super Learner Ensemble for Phishing Detection on Mobile Devices," *Scientific Reports*, vol. 15, 2025.
[11] R. Bourigue *et al.*, "Deep Learning Model for Phishing URL Detection," *Cluster Computing*, 2025.
[12] R. Alzubi *et al.*, "Feature-Based Machine Learning Approach for Detecting Phishing URLs," *Engineering, Technology and Applied Science Research*, vol. 15, 2025.
[13] "Advancing Phishing Email Detection: Comparative Study of Deep Learning Models," *Sensors*, vol. 24, 2024.
[14] "Machine Learning and Neural Networks for Phishing Detection: A Systematic Review," *Electronics*, vol. 14, 2025.
[15] S. Kavya and D. Sumathi, "Recent Advances and Emerging Methodologies in Phishing Detection," *Artificial Intelligence Review*, 2025.
[16] "Systematic Review of Deep Learning Techniques for Phishing Email Detection," *Electronics*, vol. 13, 2024.
[17] F. S. Alsubaei *et al.*, "Hybrid Deep Learning Framework for Phishing Detection Using GRU and ResNeXt," *IEEE Access*, 2024.
[18] R. Dubey *et al.*, "Ensemble Approach Using Character-Level CNN and Feature Engineering for Phishing Detection," *arXiv*, 2025.
[19] A. Al Faisal, "Deep Reinforcement Learning for Phishing Detection with Transformer-Based Features," *arXiv*, 2025.
[20] S. E. Blake, "PhishSense: AI-Powered Phishing Detection Using Large Language Model Fine-Tuning," *arXiv*, 2025.
[21] T. Ige *et al.*, "Comparative Study of Machine Learning and Deep Learning Models for Phishing Detection," 2024.
[22] J. Wilk-Jakubowski *et al.*, "AI and Vision-Based Techniques in Phishing Detection Systems," 2025.