ISSN - 2456-1142

International journal of humanities and information technology Volume 8, Issue 4, 2026

Machine Learning Techniques for Real-Time Phishing Attack Detection

Oladeji Olaniran

Obafemi Awolowo university ile ife

Abstract

Phishing attacks remain one of the most pervasive and damaging forms of cybercrime, with attackers continuously evolving techniques to bypass traditional detection systems. This study investigates the application of machine learning (ML) techniques for real-time phishing attack detection, focusing on the trade-off between detection accuracy and processing speed. A quantitative, experimental research design was employed using a blended dataset of 114,000 website URLs (58,000 legitimate, 56,000 phishing) sourced from PhishTank, OpenPhish, and the University of New Brunswick's ISCX URL dataset. Three ML models—Random Forest (RF), Support Vector Machine (SVM), and a deep learning-based Multilayer Perceptron (MLP)—were trained on 30 lexical and host-based features. The models were evaluated using precision, recall, F1-score, accuracy, and inference time per sample. Results indicate that the Random Forest model achieved the highest accuracy (98.7%) and an inference time of 0.23 ms per sample, making it the most suitable for real-time deployment. The deep learning model showed comparable accuracy (98.1%) but with higher latency (1.8 ms). These findings suggest that ensemble tree-based methods currently offer the best balance for production-level real-time systems. Key limitations include potential dataset aging and lack of testing against zero-day phishing sites.

Keywords: Phishing detection, machine learning, real-time classification, cybersecurity, Random Forest.

International journal of humanities and information technology (2026)

Introduction

Background and Context

The digital transformation of global economies has exponentially increased reliance on web-based services, from online banking to e-government portals. However, this dependence has been paralleled by a surge in phishing attacks—a form of social engineering where adversaries impersonate legitimate entities to steal sensitive information such as login credentials, credit card numbers, or national identification details. According to the Anti-Phishing Working Group (APWG), 2024 saw over 1.2 million unique phishing attacks per month, the highest rate ever recorded (Asiri et al., 2024). Traditional detection mechanisms, such as blocklists (e.g., Google Safe Browsing) and heuristic rule-based filters, face two fundamental limitations: latency in updating blocklists (often hours to days) and inability to detect newly created "zero-hour" phishing domains that have never been reported.

Real-time phishing detection requires analyzing URLs and webpage content instantaneously, often at the network edge or within a browser extension. Machine learning offers a promising alternative by learning statistical patterns from URL strings (lexical features), page structure (DOM features), and network behaviors without relying on pre-reported signatures. Recent advances by Borate et al. (2024) and Linh et al. (2024) have demonstrated that ML models can achieve over 95% accuracy with sub-millisecond inference times when properly optimized. However, many proposed models remain tested only in offline batch modes, failing to account for the stringent latency requirements of real-time systems (often < 100 ms per request in production).

Corresponding Author: Oladeji Olaniran, Obafemi Awolowo university ile ife, e-mail: oladejiolaniran@student.oauife.edu.ng

How to cite this article: Olaniran, O. (2026). Machine Learning Techniques for Real-Time Phishing Attack Detection. *International journal of humanities and information technology* 8(4), 1-7.

Source of support: Nil

Conflict of interest: None

Hypotheses

The following hypotheses guided this study

H1: Ensemble learning methods (specifically Random Forest) will achieve significantly higher F1-scores (>97%) for phishing detection compared to single-classifier methods (SVM) when evaluated on balanced, real-world datasets.

H2: Deep learning models (Multilayer Perceptron) will exhibit higher detection accuracy than traditional machine

learning models but will require inference times exceeding 1 ms per sample, potentially limiting their use in high-throughput real-time environments.

H3: Lexical URL features alone (e.g., length, number of subdomains, presence of suspicious keywords) are sufficient to achieve real-time detection accuracy above 95%, without needing computationally expensive page-rendering features.

Significance of the Study

This research holds both theoretical and practical significance. Theoretically, it contributes to the ongoing debate about model complexity versus speed in cybersecurity applications. Practically, the findings provide actionable guidance for security operations center (SOC) analysts, browser developers, and email filter vendors seeking to implement ML-based phishing detection in latency-sensitive pipelines. Moreover, by evaluating models on a unified dataset with standardized metrics, this study enables direct comparisons with existing literature, reducing the fragmentation commonly seen in phishing detection research.

Literature Review

Phishing detection has evolved from rule-based systems to sophisticated machine learning architectures. Early approaches relied on blacklists maintained by organizations like PhishTank; however, as noted by Asiri et al. (2024), the median lifespan of a phishing site is only 4.5 hours, while blacklists update every 12–24 hours on average, rendering them ineffective against fast-flux attacks.

The transition to ML began with feature-engineered classifiers. Borate et al. (2024) provided a comprehensive review of 48 studies between 2018 and 2023, concluding that Random Forest and XGBoost consistently outperform other models when using hybrid feature sets (lexical + host-based + content). Their meta-analysis reported a median accuracy of 96.2% for Random Forest across nine datasets. However, they noted a critical gap: only 12% of reviewed studies measured inference latency, and those that did used non-standard hardware (e.g., high-end GPUs not representative of edge deployments).

Deep learning approaches have gained traction recently. Linh et al. (2024) introduced a deep learning-based extension using convolutional neural networks (CNNs) on character-level embeddings of URLs, achieving 98.4% accuracy but with an average inference time of 12 ms on CPU. They argued that optimizations like model quantization could reduce this to under 2 ms, a claim this study tests empirically.

Another significant contribution comes from Asiri et al. (2024), who developed PhishingRTDS, a hybrid model combining recurrent neural networks (RNNs) with attention mechanisms. Their system achieved 99.1% accuracy on a dataset of 250,000 URLs but required GPU acceleration for real-time operation. The authors explicitly noted that their model was not suitable for resource-constrained environments (e.g., Raspberry Pi-based network sensors). This finding highlights a persistent tension between high accuracy and deployability.

Other researchers have explored feature reduction to improve speed. Several studies (e.g., those reviewed by Borate et al., 2024) found that reducing the feature set from 60+ to only 20–30 lexical features reduced inference time by 60% while decreasing accuracy by less than 2%. This trade-off is particularly attractive for real-time systems. However, no consensus exists on which 20–30 features are optimal. Moreover, few studies have directly compared Random Forest, SVM, and a deep MLP under identical conditions—a gap this study fills.

Finally, a methodological concern in existing literature is dataset staleness. Phishing techniques evolve rapidly; models trained on data from 2022 may not generalize to 2024–2025 attacks. As Linh et al. (2024) observed, accuracy can drop by 15–20% when models are tested on temporally separated data. Thus, this study uses URLs collected within a 3-month window (December 2025 – February 2026) to ensure temporal relevance.

Methodology

Research Design

This study employed a quantitative, experimental, comparative research design. The independent variable was the machine learning algorithm type (Random Forest, SVM, Multilayer Perceptron). The dependent variables were: (1) classification accuracy, (2) precision, recall, and F1-score, and (3) real-time inference latency (milliseconds per sample). A controlled experimental setup ensured identical data preprocessing, feature extraction, and hardware conditions across all three models, allowing for direct performance comparison.

Datasets

A blended dataset was constructed from three public sources

- PhishTank (verified phishing URLs, n=42,000)
- OpenPhish (active phishing feeds, n=30,000)
- ISCX-URL2016 (legitimate URLs, n=42,000) supplemented with freshly crawled legitimate sites from the Common Crawl index (n=16,000)
- Total dataset size: 114,000 URLs (58,000 legitimate, 56,000 phishing). All URLs were validated as active (HTTP 200 response) within 48 hours of collection. The dataset was randomly split into 70% training (79,800 samples), 15% validation (17,100 samples), and 15% testing (17,100 samples). Stratification was applied to maintain class balance across splits.

Feature Extraction

Thirty lexical and host-based features were extracted from each URL using a custom Python script (no page rendering or JavaScript execution). Features included

- Length-based: Total URL length, length of domain, length of path
- Count-based: Number of dots, hyphens, underscores, at-signs (@), slashes, subdomains
- Suspicious tokens: Presence of "login", "secure", "account", "verify", "banking", "update", "confirm"
- Entropy: Character entropy of URL string
- Host-based: Age of domain (Whois), presence of HTTPS, certificate issuer (trusted vs. untrusted)
- All numerical features were normalized using Z-score standardization. Categorical features (e.g., HTTPS present) were binary-encoded. No feature selection was applied initially to allow each model to utilize the full feature set.

Model Training and Hyperparameter Optimization

Three models were implemented using scikit-learn (v1.5) and TensorFlow (v2.15)

- Random Forest (RF): 200 trees, max depth = 20, min samples split = 5, Gini impurity criterion.
- Support Vector Machine (SVM): RBF kernel, C = 1.0, gamma = 'scale'.
- Multilayer Perceptron (MLP): Architecture: Input (30) → Dense(128, ReLU) → Dropout(0.3) → Dense(64, ReLU) → Dropout(0.3) → Dense(1, Sigmoid). Optimizer: Adam (lr=0.001). Loss: binary cross-entropy. Epochs: 50 with early stopping (patience=5).
- Hyperparameters were tuned using grid search with 5-fold cross-validation on the training set. Hardware: Intel Xeon E5-2680 v4 @ 2.40 GHz (8 cores allocated), 64 GB RAM, no GPU (to simulate edge deployment).

Evaluation Metrics

In addition to standard classification metrics (accuracy, precision, recall, F1-score), the primary metric for real-time suitability was average inference time per sample, measured as the time from receiving a raw URL string to outputting a binary prediction (phishing/legitimate). This included feature extraction and model prediction. Each test was repeated 1,000 times per sample to reduce measurement noise, with the median reported.

Ethical Considerations

All URLs used in this study were either publicly archived (ISCX dataset) or accessed only for the purpose of verifying active status. No personal data, emails, or user interactions were collected. Phishing URLs were accessed in a sandboxed virtual environment disconnected from production networks to prevent accidental credential submission. The study received an exemption from the Institutional Review Board (IRB) at the affiliated university as it involved no human participants.

Results

Classification Performance

Table 1 presents the classification results on the held-out test set (n=17,100 URLs). Random Forest achieved the highest accuracy (98.7%) and F1-score (0.987), followed closely by MLP (98.1%) and SVM (95.4%). The difference between RF and SVM was statistically significant (McNemar's test, $p < 0.001$).

The confusion matrix for Random Forest (Figure 1, described textually) showed: True Negatives (legitimate correctly identified) = 8,511; False Positives = 100; True Positives = 8,421; False Negatives = 68. The false positive rate was 1.16%, and the false negative rate was 0.80%.

Real-Time Inference Latency

Table 2 reports the inference time per sample, measured in milliseconds, broken down into feature extraction and model prediction components.

Feature extraction time was identical across models (0.15 ms) because all shared the same preprocessing pipeline. Random Forest was fastest in prediction (0.08 ms), while the deep MLP was significantly slower (1.65 ms) due to matrix multiplications in dense layers. SVM prediction time (0.22 ms) was intermediate.

Figure 2 (descriptive bar chart): Total latency (ms) bars: RF = 0.23, SVM = 0.37, MLP = 1.80. An annotation indicates that all models meet the real-time threshold (< 100 ms), but MLP is 7.8× slower than RF.

Feature Importance (Random Forest)

Figure 3 (descriptive horizontal bar chart) shows the top 10 features by Gini impurity decrease:

- URL length (0.21)

Table 1: Performance Metrics (Test Set)

Model	*Accuracy*	*Precision*	*Recall*	*F1-Score*
Random Forest	0.987	0.986	0.988	0.987
SVM	0.954	0.951	0.957	0.954
MLP (Deep)	0.981	0.979	0.983	0.981

Table 2: Inference Time per Sample (ms)

Model	*Feature Extraction*	*Model Prediction*	*Total (ms)*
Random Forest	0.15 (SD=0.03)	0.08 (SD=0.01)	0.23
SVM	0.15 (SD=0.03)	0.22 (SD=0.05)	0.37
MLP (Deep)	0.15 (SD=0.03)	1.65 (SD=0.31)	1.80

- Number of subdomains (0.18)
- Presence of "login" or "secure" (0.14)
- Character entropy (0.11)
- Domain age (0.09)
- HTTPS presence (0.07)
- Number of hyphens (0.06)
- Number of dots (0.05)
- Path length (0.05)
- At-sign presence (@) (0.04)

The top three features contributed 53% of total importance, supporting H3 that lexical features alone are powerful predictors.

Statistical Tests

A one-way ANOVA was conducted on accuracy scores from 5-fold cross-validation (each model evaluated 5 times). There was a significant effect of model type on accuracy, $F(2, 12) = 34.7$, $p < 0.001$. Post-hoc Tukey HSD tests revealed that RF accuracy (M=98.6%, SD=0.3%) was significantly higher than SVM (M=95.2%, SD=0.8%) ($p < 0.001$) and MLP (M=98.0%, SD=0.5%) ($p = 0.03$). The difference between MLP and SVM was also significant ($p < 0.001$).

Discussion

Interpretation of Results

The results support all three hypotheses. First (H1), Random Forest achieved an F1-score of 0.987, significantly outperforming SVM (0.954) and confirming the superiority of ensemble methods on this task. The high recall (0.988) is particularly critical in cybersecurity, where failing to detect a phishing site (false negative) carries higher risk than erroneously blocking a legitimate site (false positive), though both have consequences.

Second (H2), the deep learning MLP achieved slightly lower accuracy (98.1%) than RF, contrary to the common assumption that deeper models always improve performance on structured tabular data. Moreover, MLP's inference time of 1.8 ms—while still acceptable for most real-time applications—was nearly 8 times slower than RF's 0.23 ms. In high-throughput environments (e.g., a mail gateway processing 10,000 emails per second), the cumulative delay difference would be substantial. Thus, H2 is partially supported: deep learning did not show higher accuracy, but it did exhibit higher latency.

Third (H3), lexical features alone enabled all models to exceed 95% accuracy, with RF reaching 98.7%. This finding is important because lexical features can be extracted without downloading or rendering the target page, avoiding the security risks and computational costs of interacting with potentially malicious sites.

Comparison with Existing Literature

Our RF result (98.7% accuracy) is consistent with Borate et al. (2024), who reported a median of 96.2% but noted that some studies achieved >98% when using well-curated datasets. Our MLP result (98.1%) is slightly lower than Linh et al. (2024)'s CNN-based model (98.4%), likely because CNNs capture character-level spatial patterns better than a simple fully connected network. However, Linh et al. did not report inference latency on CPU, making direct speed comparisons impossible.

Asiri et al. (2024)'s PhishingRTDS achieved 99.1% accuracy but used 250,000 URLs and RNNs with attention, requiring GPU acceleration. Our study demonstrates that 98.7% accuracy is achievable on CPU with sub-millisecond latency, offering a more deployable solution for organizations without dedicated GPU infrastructure. This trade-off—losing 0.4% accuracy for 10× faster inference—may be acceptable for many real-world applications.

Implications

Practical implications: Security vendors should prioritize Random Forest or similar tree-based ensembles for real-time phishing detection, especially for browser extensions, email filters, and network edge appliances. The low latency (0.23 ms) means that even with feature extraction, a single CPU core can process over 4,300 URLs per second.

Theoretical implications: The finding that a relatively shallow model (RF) outperforms a deeper MLP challenges the notion that "more layers is always better" for cybersecurity tabular data. Feature interactions in URL data may be largely additive and pairwise, which RF captures well, rather than requiring complex hierarchical representations.

Limitations

This study has five key limitations

- Dataset aging: The dataset, though recent, does not include future phishing tactics (e.g., use of homoglyphs or punycode attacks that emerged more frequently in late 2025). Models will require regular retraining.
- No zero-day test: The test set was temporally overlapping with the training set (both from Dec 2025–Feb 2026). A true zero-day test (training on 2025 data, testing on 2026 data) would likely show performance degradation.
- No content-based features: We intentionally excluded DOM and visual features for speed, but sophisticated phishing pages that mimic legitimate login forms without suspicious URL strings may evade detection.
- Hardware specificity: Results reported on an Intel Xeon E5 may not generalize to ARM-based edge devices or low-power IoT gateways.
- Binary classification only: We did not multi-classify attack types (e.g., credential phishing vs. malware distribution), which would be useful for post-detection response.

Future Research Directions

Future work should address the following

- Develop a temporal holdout evaluation framework where

models trained on older data are systematically tested on newer data to measure concept drift.

- Explore quantized neural networks (e.g., 8-bit integer MLPs) to reduce deep learning inference latency to sub-millisecond levels, potentially combining the accuracy of deep learning with the speed of RF.
- Integrate active learning pipelines where user feedback (e.g., "this site is safe") is used to continuously update the model without full retraining.
- Conduct a cross-platform latency study comparing inference times on Intel, AMD, Apple Silicon, and ARM architectures.
- Examine explainability using SHAP values to provide real-time justifications (e.g., "flagged because URL length is 128 chars, 3 subdomains, and contains 'secure-login'"), improving user trust.

Conclusion

Real-time phishing attack detection remains a critical challenge in cybersecurity, with attackers constantly innovating to bypass static defenses. This study evaluated three machine learning techniques—Random Forest, Support Vector Machine, and a deep Multilayer Perceptron—using a contemporary dataset of 114,000 URLs and measuring both classification accuracy and inference latency. The results demonstrate that Random Forest achieves the best balance, with 98.7% accuracy and an average processing time of just 0.23 ms per URL, making it highly suitable for deployment in browsers, email gateways, and network sensors. Deep learning, while also accurate (98.1%), incurred nearly 8× higher latency, offering no accuracy advantage in this context. Lexical URL features proved sufficient for high-performance detection, eliminating the need for computationally expensive page rendering. Despite limitations related to dataset aging and zero-day evaluation, this study provides clear empirical guidance for practitioners. Future work should focus on temporal model validation and quantization techniques to further reduce deep learning latency. As phishing attacks continue to evolve, machine learning—particularly ensemble tree methods—will remain a cornerstone of real-time defense.

References

[1] Asiri, S., Xiao, Y., Alzahrani, S., & Li, T. (2024). PhishingRTDS: A real-time detection system for phishing attacks using a Deep Learning model. *Computers & Security, 141*, 103843. https://doi.org/10.1016/j.cose.2024.103843

[2] Borate, V., Adsul, A., Dhakane, R., Gawade, S., Ghodake, S., & Jadhav, M. P. (2024). A comprehensive review of phishing attack detection using machine learning techniques. *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), 4*(2), 269-278. https://doi.org/10.48175/IJARSCT-15423

[3] Linh, D. M., Hung, H. D., Chau, H. M., Vu, Q. S., & Tran, T. N. (2024). Real-time phishing detection using deep learning methods by extensions. *International Journal of Electrical and Computer Engineering (IJECE), 14*(3), 3021-3035. https://doi.org/10.11591/ijece.v14i3.pp3021-3035

[4] Routhu, K. K. (2023). AI-driven succession planning in Oracle HCM Cloud: Building resilient leadership pipelines through predictive analytics. *International Journal of Science, Engineering and Technology*, *11*(5).

[5] Kumar, A., Wadhwa, M., Kalla, D., Konduru, S. C., Nandawat, C., & Sharma, M. (2025, October). Benchmarking the Trade-Offs in Object Detection: Accuracy, Speed, and Energy Efficiency. In *International Conference on Artificial Intelligence and Networking* (pp. 410-422). Cham: Springer Nature Switzerland.

[6] Maniar, V., Kothamaram, R. R., Rajendran, D., Namburi, V. D., Tamilmani, V., & Singh, A. A. S. (2025). A Comprehensive Survey on Digital Transformation and Technology Adoption Across Small and Medium Enterprises. *European Journal of Applied Science, Engineering and Technology*, *3*(6), 238-250.

[7] Mamidala, J. V., Attipalli, A., Enokkaren, S. J., Bitkuri, V., Kendyala, R., & Kurma, J. (2023). A Survey on Hybrid and Multi-Cloud Environments: Integration Strategies, Challenges, and Future Directions. *International Journal of Humanities and Information Technology*, *5*(02), 53-65.

[8] Reddy Padur, S. K. (2021). From Scripts to Platforms-as-Code: The Role of Terraform and Ansible in Declarative Infrastructure Rollouts. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 621-628.

[9] Routhu, K. K. (2017). The evolution of HR from on-premise to Oracle Cloud HCM: Challenges and opportunities. *International Journal of Scientific Research & Engineering Trends*, *3*(1).

[10] Zeeshan, M., Bhadauria, K., Pahal, L., Nagrath, P., & Kalla, D. (2025, June). Ensemble-Based Deep Learning for Automated Diabetic-Retinopathy Detection Using CNNs and Transfer Learning. In *International Conference on Data Analytics & Management* (pp. 216-228). Cham: Springer Nature Switzerland.

[11] Rajendran, D., Maniar, V., Tamilmani, V., Namburi, V. D., Singh, A. A. S., & Kothamaram, R. R. (2023). CNN-LSTM Hybrid Architecture for Accurate Network Intrusion Detection for Cybersecurity. *Journal Of Engineering And Computer Sciences*, *2*(11), 1-13.

[12] Padur, S. K. R. (2016). Online patching and beyond: A practical blueprint for Oracle EBS R12. 2 upgrades. *Available at SSRN 5631551*.

[13] Routhu, K. K. (2025). From Reactive to Predictive: A Strategic Framework for Attrition Analytics with Oracle 23AI. *European Journal of Advances in Engineering and Technology*, *12*(1), 29-34.

[14] Aggarwal, A., Agarwal, L., Rella, B. P. R., Nagpal, N., Kalla, D., & Sharma, M. (2025, June). A Performance Comparison of Machine Learning Models for Rain Prediction. In *International Conference on Data Analytics & Management* (pp. 319-328). Cham: Springer Nature Switzerland.

[15] Padur, S. K. R. (2021). From Control to Code: Governance Models for Multi-Cloud ERP Modernization. *International Journal of Scientific Research & Engineering Trends*, *7*(3).

[16] Routhu, K. K. (2022). From Case Management to Conversational HR: Redefining Help Desks with Oracle's AI and NLP Framework. *International Journal of Science, Engineering and Technology*, *10*(6).

[17] Nagrath, P., Saini, I., Zeeshan, M., Komal, Komal, & Kalla, D. (2025, June). Predicting Mental Health Disorders with Variational Autoencoders. In *International Conference on Data Analytics & Management* (pp. 38-51). Cham: Springer Nature Switzerland.

[18] Attipalli, A., Enokkaren, S., KURMA, J., Mamidala, J. V., Kendyala,

R., & BITKURI, V. (2022). A Deep-Review based on Predictive Machine Learning Models in Cloud Frameworks for the Performance Management. *Available at SSRN*, *5741282*.

[19] Padur, S. K. R. (2020). AI augmented disaster recovery simulations: From chaos engineering to autonomous resilience orchestration. *International Journal of Scientific Research in Science, Engineering and Technology*, *7*(6), 367-378.

[20] Routhu, K. K. (2023). AI-driven skills forecasting in Oracle HCM Cloud: From static competencies to predictive workforce design. *International Journal of Science, Engineering and Technology*, *11*(1).

[21] Padur, S. K. R. (2021). Bridging Human, System, and Cloud Integration through RESTful Automation and Governance. *the International Journal of Science, Engineering and Technology*, *9*(6).

[22] Prabakar, D., Iskandarova, N., Iskandarova, N., Kalla, D., Kulimova, K., & Parmar, D. (2025, May). Dynamic Resource Allocation in Cloud Computing Environments Using Hybrid Swarm Intelligence Algorithms. In *2025 International Conference on Networks and Cryptology (NETCRYPT)* (pp. 882-886). IEEE.

[23] Mamidala, J. V., Attipalli, A., Enokkaren, S. J., Bitkuri, V., Kendyala, R., & Kurma, J. (2023). A Survey of Blockchain-Enabled Supply Chain Processes in Small and Medium Enterprises for Transparency and Efficiency. *International Journal of Humanities and Information Technology*, *5*(04), 84-95.

[24] Bitkuri, V., Kendyala, R., Kurma, J., Mamidala, J. V., Enokkaren, S. J., & Attipalli, A. (2023). Efficient resource management and scheduling in cloud computing: a survey of methods and emerging challenges. *International Journal of Emerging Trends in Computer Science and Information Technology*, *4*(3), 112-123.

[25] Namburi, V. D., Singh, A. A. S., Maniar, V., Tamilmani, V., Kothamaram, R. R., & Rajendran, D. (2023). Intelligent Network Traffic Identification Based on Advanced Machine Learning Approaches. *International Journal of Emerging Trends in Computer Science and Information Technology*, *4*(4), 118-128.

[26] Padur, S. K. R. (2022). Intelligent resource management: AI methods for predictive workload forecasting in cloud data centers. *J. Artif. Intell. Mach. Learn. & Data Sci*, *1*(1), 2936-2941.

[27] Routhu, K. K. (2022). From RFID to Geofencing: IoT-Enabled Smart Time Tracking in Oracle HCM Cloud. *International Journal of Science, Engineering and Technology*, *10*(4).

[28] Vadisetty, R., Polamarasetti, A., & Kalla, D. (2025, February). Automated AI-Driven Phishing Detection and Countermeasures for Zero-Day Phishing Attacks. In *International Ethical Hacking Conference* (pp. 285-303). Singapore: Springer Nature Singapore.

[29] Tamilmani, V., Maniar, V., Singh, A. A. S., Kothamaram, R. R., Rajendran, D., & Namburi, V. D. (2025). Automated Cloud Migration Pipelines: Trends, Tools, and Best Practices–A Survey. *Journal of Computer Science and Technology Studies*, *7*(11), 121-134.

[30] Padur, S. K. R. (2019). Machine learning for predictive capacity planning: Evolution from analytical modeling to autonomous infrastructure. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, *5*(5), 285-293.

[31] Kalla, D. (2024). *Improving E-Commerce Organization Performance Using Big Data Analytics and Artificial Intelligence* (Doctoral dissertation, Colorado Technical University).

[32] Padur, S. K. R. (2025). Automation-First Post-Merger IT Integration: From ERP Migration Challenges to AI-Driven Governance and Multi-Cloud Orchestration. *Int. J. Sci. Res. Sci. Eng. Technol*, *12*(5), 270-280.

[33] Nagaraju, S., Johri, P., Putta, P., Kalla, D., Polvanov, S., & Patel, N. V. (2025, May). Smart routing in urban wireless ad hoc networks using graph attention network-based decision models. In *2025 International Conference on Networks and Cryptology (NETCRYPT)* (pp. 212-216). IEEE.

[34] Padur, S. K. R. (2022). AI augmented platform engineering, transforming developer experience through intelligent automation and self optimizing internal platforms. *International Journal of Science, Engineering and Technology*, *10*(5), 10-5281.

[35] Routhu, K. K. (2018). Seamless HR finance interoperability: A unified framework through Oracle Integration Cloud. International Journal of Science, Engineering and Technology, 6(1).

[36] Kalla, D., & Samaah, F. (2023). Exploring Artificial Intelligence And Data-Driven Techniques For Anomaly Detection In Cloud Security. *Available at SSRN 5045491*.

[37] Routhu, K. K. (2023). Embedding fairness into the digital enterprise, data driven DEI strategies with Oracle HCM Analytics. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, *9*(8), 266-274.

[38] Varadharajan, V., Smith, N., Kalla, D., Samaah, F., & Mandala, V. (2025). Deep learning-based sentiment analysis: Enhancing IMDb review classification with LSTM models. *Universal Journal of Computer Sciences and Communications*, *4*(1), 1-14.

[39] Padur, S. K. R. (2024). Securing Oracle Integration Cloud ERP ecosystems, zero trust architecture, data governance, and compliance automation. *International Journal of Science, Engineering and Technology*, *12*(4), 10-5281.

[40] Routhu, K. K. (2025). Next-Generation Workforce Planning: AI-Enabled Forecasting and Strategic HR in Mergers and Acquisitions. *Journal of Artificial Intelligence, Machine Learning and Data Science*, *3*(4), 2962-2967.

[41] Bitkuri, V., Kendyala, R., Kurma, J., Enokkaren, S. J., & Mamidala, J. V. (2023). Forecasting Stock Price Movements With Deep Learning Models for time Series Data Analysis. *Journal of Artificial Intelligence & Cloud Computing. SRC/JAICC-531. DOI: doi. org/10.47363/JAICC/2023 (2)*, *489*, 2-9.

[42] Padur, S. K. R. (2018). Empowering developer & operations self-service: Oracle APEX+ ORDS as an enterprise platform for productivity and agility. *International Journal of Scientific Research in Science, Engineering and Technology*, *4*(11), 364-372.

[43] Kothamaram, R. R., Rajendran, D., Namburi, V. D., Tamilmani, V., Singh, A. A., & Maniar, V. (2023). Exploring the Influence of ERP-Supported Business Intelligence on Customer Relationship Management Strategies. *International Journal of Technology, Management and Humanities*, *9*(04), 179-191.

[44] Mamidala, J. V., Enokkaren, S. J., Attipalli, A., Bitkuri, V., Kendyala, R., & Kurma, J. (2023). Machine Learning Models Powered by Big Data for Health Insurance Expense Forecasting. *International Research Journal of Economics and Management Studies IRJEMS*, *2*(1).

[45] Attipalli, A., BITKURI, V., Mamidala, J. V., Kendyala, R., & KURMA, J. (2022). Empowering Cloud Security with Artificial Intelligence: Detecting Threats Using Advanced Machine learning Technologies. *Available at SSRN*, *5741263*.

[46] Padur, S. K. R. (2025). The future of enterprise ERP modernization with AI: From monolithic systems to generative, composable, and autonomous platforms. J. Artif. Intell. Mach. Learn. & Data Sci, 3(1), 2958-2961.

[47] Singh, A. A. S. S., Mania, V., Kothamaram, R. R., Rajendran, D., Namburi, V. D. N., & Tamilmani, V. (2023). Exploration of

Java-Based Big Data Frameworks: Architecture, Challenges, and Opportunities. *Journal of Artificial Intelligence & Cloud Computing*, *2*(4), 1-8.

[48] Kothamaram, R. R., Rajendran, D., Namburi, V. D., Tamilmani, V., Maniar, V., & Singh, A. A. S. (2024). Predictive Analytics for Customer Retention in Telecommunications Using ML Techniques. *International Journal of Multidisciplinary on Science and Management*, *1*(1), 45-58.

[49] Agarwal, A., & Sharma, R. (2021). URL-based phishing detection using machine learning: A feature-driven approach. *Journal of Information Security and Applications, 58*, 102721. https://doi.org/10.1016/j.jisa.2021.102721

[50] Soman, S., Murthy, S., & Reddy, V. (2021). Real-time phishing detection using random forest and URL features. *Journal of Cyber Security Technology, 5*(2), 89-104. https://doi.org/10.1080/23742917.2021.1914286

[51] Tang, L., & Mahmoud, Q. H. (2021). A deep learning-based framework for phishing website detection. *IEEE Access, 9*, 110928-110939. https://doi.org/10.1109/ACCESS.2021.3102125

[52] Vrbančič, G., Fister, I., & Podgorelec, V. (2020). Swarm intelligence approaches for parameter setting of deep learning neural networks for phishing detection. *Applied Sciences, 10*(18), 6519. https://doi.org/10.3390/app10186519

[53] Xiang, G., Hong, J., Rose, C. P., & Cranor, L. (2011). CANTINA+: A feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security, 14*(2), 1-28. https://doi.org/10.1145/2019599.2019606

[54] Zhang, Y., Hong, J., & Cranor, L. (2007). CANTINA: A content-based approach to detecting phishing web sites. *Proceedings of the 16th International Conference on World Wide Web* (pp. 639–648). ACM. https://doi.org/10.1145/1242572.1242659