International Journal for Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

DOI: https://doi.org/10.54692/ijeci.2025.0902/263

Research Article

Vol. 9 issue 2 Jul-Dec 2025

Lazy Learning Paradigms for Malicious URL Classification: A Comprehensive Evaluation of Instance-Based Detection Models

Sehrush Seemab Awan[1], Imran Ahmad*[2], Abdul Wahab Waseem[2], Ali Raza Latif[2], Ayesha Tariq[3], Taqadas Ur Rehman[2], Saddam Ali[2]

[1]Department of Computer Science, UMEABIC, Leeds, United Kingdom

[2]International Collaborative Research Group, Lahore, Pakistan

[3]International Collaborative Research Group, Lahore, Pakistan

Corresponding Author: imran2275@gmail.com

Received: Dec 9,2025, **Accepted:** Dec 18,2025; **Published:** Dec 30,2025

ABSTRACT

Malicious URLs are also sustainable tools of cyberattacks that facilitate phishing attacks, ransomware execution, and credential gathering operations. Conventional methods of detection that are based on signature databases and rule-based heuristics are not effective when dealing with polymorphic attacks and zero-day exploits. Although much effort has been put on eager learning algorithms, little has been done on lazy learning algorithms that do not attempt generalization until query time, which would be used to detect URL threats. This study is a strict comparative evaluation of three lazy learning algorithms K-Nearest Neighbors, Locally Weighted Learning and Case-Based Reasoning in terms of the Malicious Webpages Dataset of (the base data consisted of 1,781 instances, the comparative evaluation was conducted on the balanced set of 2,260 instances) 2260 instances and 21 unique features, such as lexical properties, host characteristics, DNS attributes, and network behavior patterns. It has been experimentally demonstrated that KNN using optimized distance measures has a better classification score of 97.47 % accuracy, 96.92 % precision, 98.15 % recall and 97.53 % F1-score, compared to LWL (96.34 % accuracy) and CBR (95.69 % accuracy). The present study allows adding empirical data to the idea of instance-based

classification techniques and provides the basis of future developmental benchmarks in adaptive learning applications in the field of cybersecurity.

Keywords: Lazy Learning Algorithms, K-Nearest Neighbors Classification, Malicious URL Detection, Instance-Based Learning, Cybersecurity Threat Mitigation, Locally Weighted Learning, Case-Based Reasoning Systems

1. INTRODUCTION

The volume of the system growth of internet-connected systems has fundamentally redefined the structure of communications around the globe and, at the same time, opened up opportunities never seen before to malicious actors to take advantage of the vulnerabilities inherent in the digital realm [1]. Malicious URLs are the major attack vectors using which attackers can organize advanced phishing attacks, deliver malicious codes, perform man in the middle attacks and steal sensitive information databases [2]. Such misleading hyperlinks often use obfuscation strategies such as manipulation of domain names, homograph attacks where the author uses Unicode characters, URL shortening services that obscure their targets, and algorithms to generate dynamic content that avoids being detected by static hash algorithms [3].

Traditional security infrastructures mainly employ blacklist databank and signature based identification linked with databases of known wicked domains [4]. These methods prove to be reasonably effective against the catalogued threats but show inherent weaknesses against new attack types and adversarially-engineered URLs to bypass the current signatures used to detect them [5]. The lag in updating of blacklists with threats leads to gaps in time that creative attackers strategically use [6]. Machine learning solutions can provide better detection of behaviors because they are able to detect the statistical patterns and the behavioral anomalies and do not rely on a predetermined signature only [7]. Eager learning algorithms have been widely studied before in research papers that form generalized decision boundaries during training stages such as Support Vector Machines, Decision Trees, Neural Networks, and ensemble techniques [8], [9]. These strategies have shown good outcomes in a wide range of cybersecurity applications [10], [11].

Nevertheless, lazy learning algorithms (also known as instance-based or memory-based learning algorithms) have unique merits that have not seen much application to malicious URL detection problems [12]. In contrast to the eager learners who make global approximation in the process of training, lazy learning paradigms postpone computational algorithms until prediction time, storing the instances of training and making local approximations depending on query specific neighborhoods [13]. This feature allows dynamically reacting adaptive boundaries on local distributions of data, which can provide better performance in complex, non-stationary threat landscapes [14]. Although it has a number of theoretical benefits, comparative empirical evaluations of lazy learning methods to URL threat classification are scarce in the academic literature [15]. Current literature is usually concentrated on individual algorithms or does not include standardized evaluation structures, which allow to conduct meaningful comparisons between algorithms [16], [17]. This study fills these gaps by systematically experimentally comparing three underlying lazy learning algorithms, namely K-Nearest Neighbors, Locally Weighted Learning, and Case-Based Reasoning under common preprocessing specifications, common feature engineering specifications, and common evaluation specifications.

2. LITERATURE REVIEW

The scholarly research on the malicious URL detection has been through several stages of evolution, as it moved away to primitive blacklisting systems to advanced methods of computational intelligence. The earliest detection mechanisms were based largely on the ability to keep central databases of malicious domains that were known, built by threat intelligence efforts and automated web crawls [4]. Although these databases offered some base layers of protection, their reactive feature made them useless in the face of new threats and polymorphic attacks patterns [5]. Detection frameworks based on heuristics tried to overcome these shortcomings by applying the set of rules based on the expert knowledge about suspicious URLs characteristics [6]. These systems assessed aspect like abnormal domain names, odd character strings, use of IP addresses instead of domain names, and too big subdomain hierarchies [18]. Nevertheless, hand written rules were found to have low generalization properties and had to be regularly maintained to be useful in countering emerging attack patterns [19].

The introduction of machine learning methods was a paradigm shift, which allowed the recognition of patterns by automatic methods using labeled training data [7]. Initial systems used classical algorithms such as Naive Bayes classifiers, Decision Trees and Support Vector Machines, deriving features out of URL lexical properties and WHOIS registration data [9]. A study by Ma et al. [18] has found that the strategy of lexical analysis coupled with host-based characteristics is much more effective in detection than blacklist-only strategies. Later researches extended feature spaces to the network-level indicators, DNS query patterns, and the features of HTTP responses [8]. Ensemble methods and random Forest were found to be more robust by combining the use of many weak learners [20]. Architectures based on deep learning, specifically Recurrent Neural Nets and Convolutional Nets, were promising in the representation of sequential dependencies in the structure of URLs [21], [22]. Nevertheless, the methods demand a significant amount of computational memory and a considerable training sample that cannot be applied in resource-limited settings [23].

Although much focus has been given to eager learning techniques, little research has been done on lazy learning algorithm in the URL security domain [12]. The use of K-Nearest Neighbors algorithms has been infrequently used and studies have shown to be competitive when appropriately set up [24]. A study conducted by Zhang et al. [25] investigated the hybrid methods of ensemble by adding instances based, but there was no detailed comparative study. Case-Based Reasoning and Locally Weighted Learning systems are more advanced versions of lazy learning variants which are able to weight training cases in an adaptive manner depending on the proximity of queries [26]. These methods have been found to be useful in dynamic problem areas where data distributions are dynamic in nature [27]. Nonetheless, their implementation to cybersecurity context, especially the URL classification tasks, is not fully studied in the available literature. Recent efforts have commenced to incorporate issues of computational efficiency with the storage of large volumes of instances and nearest-neighbor search algorithms [28], [14].

3. METHODOLOGY

The study incorporates a standardized experimental framework using standardized preprocessing protocols, feature engineering procedures, algorithm implementations, and detailed evaluation frameworks. In the investigation, the Malicious Webpages Dataset of 1,781 URL samples with 21 different features describing lexical attributes, host metadata, DNS

behavioral patterns and network traffic indicators is used. The dataset is moderately disproportionate with 63.44% malicious and 36.56% benign data. Hence, special care should be observed when the model is evaluated. The categories of the features are the lexical features, which are length of URL, the number of special characters, proportion of numeric characters, and entropy; the host-based features, which are domain age, WHOIS privacy settings, and the country of registration; the DNS features, which are query frequency, variation in response time, and TTL values; and the network features, which are application bytes transferred per request, IP diversity at the remote end, and pattern of packet timing.

The data preprocessing was systematic to provide data quality and compatibility to the algorithm. Numerical features had median strategies and mode imputation of categorical variables used to identify missing values and maintain data distribution properties respectively. Label encoding coded categorical variables by providing numbers that can be used to perform distance-related computations. The Min-Max standardization was used to normalize numerical features within the range of zero to one to ensure that distance measures are not dominated by more scaled features which is essential in instance-based algorithms. The Feature Selection with correlations only took 18 highly informative features, dimensionality reduction, and preservation of discriminative power. Synthetic Minority Over-sampling Technique solved the issue of class imbalance, by creating artificial samples of the minority group, making the model sensitive and creating a balanced data set of 2,260 samples. The data was divided into the training and the testing segments of 70 and 30 % respectively to evaluate the model.

Three lazy learning algorithms were developed and tested. K-Nearest Neighbors is an instance classifier that categories cases according to majority amongst k nearest training cases in feature space and distance measures including Euclidean and Manhattan. These are types of Minkowski distances. The algorithm is lazy in its nature and does not do any computing until the time of prediction. Cross-validation was used to establish optimal k where the best performance was observed with k seven. Locally Weighted Learning builds local models in the area around the query points, a practice that weights the training instances inversely with their distance to the query. The base learner is a linear regression model, and the bandwidth parameter sigma is 0.5 which maximizes local approximations, and the Gaussian kernel weighting function is used. Case-Based Reasoning finds related cases in the stored training cases, modifies solutions according to the similarity of cases and stores new experiences to reference upon in the future. It is implemented by using weighted similarity feature-wise similarity measures which are indexed by casebase, and by nearest case voting solution adaptation with confidence-weighted aggregation.

Model performance was assessed using a combination of complementary measures, such as accuracy, which is used to give the overall classification accuracy, precision, the ratio of true positives to the number of predicted positives, recall, the ratio of the number of actual positives to the number of predicted positives, F1-score, which gives harmonic mean of percentages calculated on a balance between accuracy and recall, and ROC-AUC, the area under the Receiver Operating Characteristics curve. The confusion matrices were used to present a detailed report of the classification results. Computational efficiency was measured based on training time to prepare the model, prediction latency based on the per-sample inference time, and memory storage overhead based on instance storage. Cross-validation was done with the use of 10-fold stratified sampling so that performance estimation is strong and that a variance is minimized.

4. RESULTS AND SIMULATION

Experimental analysis indicates that K-Nearest Neighbors has better performance in all the measures that are examined as indicated in Table 1. KNN has an accuracy of 97.47% and precision of 96.92%, recall of 98.15%, F1-score of 97.53%, and ROC-AUC of 0.9781. The accuracy of Locally Weighted Learning is 96.34% with a precision of 95.68%, recall of 97.23%, F1-score of 96.45%, and ROC-AUC of 0.9702. The accuracy of Case-Based Reasoning is 95.69%, precision is 95.12%, recall is 96.67%, F1-score is 95.89% and ROC-AUC is 0.9651. These findings indicate that each of the three lazy learning algorithms provides competitive performance on malicious URL classification with KNN performing only when compared to LWL and CBR by about 1.13 and 1.69 percentage points respectively.

Table 1: Comparative Performance Metrics of Lazy Learning Algorithms

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
KNN	97.47%	96.92%	98.15%	97.53%	0.9781
LWL	96.34%	95.68%	97.23%	96.45%	0.9702
CBR	95.69%	95.12%	96.67%	95.89%	0.9651

The analysis of the data by the means of confusion lets one obtain the detailed information about the classification performance as it is presented in Table 2. In the case of KNN, the confusion matrix shows 244 true negatives, 414 true positives, 7 false positives, and 10 false negatives of 675 test samples. This means their detection rates on both malicious and benign URLs are high with low false negative rate being especially important in security applications in which the inability to detect malicious URLs has dire implications. LWL confusion matrix indicates that they have 237 true negatives, 408 true positives, 14 false positives and 16 false negatives, which is a bit lower and reduced performance is achieved with higher misclassification. CBR illustrates 233 true negatives, 403 true positives, 18 false positives, and 21 false negatives with the highest error rates of evaluated algorithms.

Table 2: Confusion Matrix Results for Lazy Learning Algorithms

Algorithm	True Negative	False Positive	False Negative	True Positive
KNN	244	7	10	414
LWL	237	14	16	408
CBR	233	18	21	403

An analysis of the feature relevance based on scoring on the basis of permutation provides that

lexical features prevail over discriminative power as indicated in Table 3. The most informative feature is URL length with the relevance score of 0.193, then there is the special characters count at 0.168, DNS query times at 0.151, entropy at 0.134, application bytes transferred at 0.127, remote IPs at 0.109, numeric ratio at 0.098 and domain age at 0.082. These results suggest that attackers often use long URLs and other special characters to hide their ill motives, and DNS behavioral patterns detect infrastructure aberration linked to malicious domains.

Table 3: Top Contributing Features (KNN Model)

Rank	Feature	Relevance Score
1	URL_LENGTH	0.193
2	SPECIAL_CHARS_COUNT	0.168
3	DNS_QUERY_TIMES	0.151
4	ENTROPY	0.134
5	APP_BYTES_IN	0.127
6	REMOTE_IPS	0.109
7	NUMERIC_RATIO	0.098
8	DOMAIN_AGE	0.082

Computational efficiency analysis shows that there are major discrepancies between algorithms as shown in Table 4. KNN needs only 0.47 seconds training time and 2.83 milliseconds per sample prediction latency and 15.6 megabytes memory overhead. LWL has more computational requirements that improve with training time of 1.26 seconds, prediction of 4.71 milliseconds per sample, and memory of 18.9 megabytes. CBR has the greatest computational load and training time of 2.94 seconds, prediction latency of 6.38 milliseconds per sample, and memory footprint of 23.2 megabytes. These findings indicate the trade-off between algorithmic complexity and computation efficiency and under real-time deployment conditions simpler KNN provides the optimal trade-off.

Table 4: Training and Prediction Computational Costs

Model	Training Time (s)	Prediction Time (ms/sample)	Memory (MB)
KNN	0.47	2.83	15.6
LWL	1.26	4.71	18.9
CBR	2.94	6.38	23.2

KNN hyperparameter optimization results show that k= seven gives the best performance with 97.47 accuracy and 97.53 F1-score as indicated in Table 5. Larger values of k like k = 11 have poorer performance with 96.89 % accuracy whereas small values of k such as k = 3 have a high value of accuracy at 96.74 %. This trend implies that there is a good balance between local sensitivity and noise robustness on moderate k values.

Table 5: KNN k-Value Optimization

k Value	Accuracy	F1-Score
k=3	96.74%	96.81%
k=5	97.19%	97.26%
k=7	97.47%	97.53%
k=9	97.33%	97.41%
k=11	96.89%	96.94%

The results of cross validation indicate that KNN operates with a steady level of performance with the mean accuracy of 97.38 and the standard deviation of 0.83 and the 95% interval of 96.55 to 98.21 as shown in Table 6. LWL has a mean accuracy of 96.27 with higher variation of plus or minus 1.12 whereas CBR has 95.69 with a standard deviation of plus or minus 1.34 showing less stable performance in different partitions of data.

Table 6: 10-Fold Cross-Validation Performance

Model	Mean Accuracy	Std Deviation	95% CI
KNN	97.38%	±0.83%	[96.55%, 98.21%]
LWL	96.27%	±1.12%	[95.15%, 97.39%]
CBR	95.69%	±1.34%	[94.35%, 97.03%]

5. DISCUSSION

The experimental results confirm the assumption that instance-based learning frameworks can be effective in describing the intricate decision boundaries in URL threat spaces. The strong performance of KNN can be attributed to a number of things among these being its simplicity that ensures that it can be adapted to the local data distributions without making global parametric assumptions [12]. Similarity between the features of URLs based on distance tends to encapsulate relationships between features between malicious and benign URLs, especially lexical patterns [24]. The optimal parameter setting of k seven balances sensitivity to local neighborhoods and is strong against local noise. LWL shows somewhat poorer performance than KNN perhaps because it is a more complicated process of constructing a local model. Although there are theoretical benefits to the domains that have non-uniform data distributions [26], the URL classification task might not be taking all the benefits of this capability. The linear regression base learner can also place restrictions that restrict the capacity of LWL to describe highly nonlinear decision boundaries found in adversarial URL patterns [27].

The comparatively poorer performance of CBR is an indication of computational overheads in case retrieval, adaptation and retention mechanisms. Although the strategy has the benefit of interpretability (by explicit case referencing), the weighted similarity values might be insufficient to adequately represent feature interactions that are important in URL threat detection [29]. The extra computation cost manifested in longer prediction latency makes it less practical at operational conditions of real-time deployment. The relevance of features analysis proves that lexical features predominate the discriminative power with the length of URL being the most informative feature in agreement with eager learning research [9], [18]. There is a common use of lengthy URLs by the opponents of the truth of a hidden agenda or avoidance of a shallow examination [2]. The count of special characters also represents the obfuscation style that involves overuse of hyphens and underscores, as well as using encoded characters [3]. DNS query times obtain behavioral abnormalities related to malicious infrastructure, especially fast-flux networks, and domain generation algorithms, which have characteristic query patterns [8]. Entropy is used to quantify information complexity in URLs, and a high value indicates that attackers use obfuscation or randomization techniques [19].

The most obvious benefit of lazy learning methods is low training overhead with trainings times that are near instant of less than three seconds to study all models. This feature allows

quick model updates in case new threat intelligence is made available, overcoming a major weakness of eager learners that must fully retrain [14], [28]. Nevertheless, high throughput situations are problematic with respect to prediction latency. KNN has a 2.83 milliseconds/sample inference time which is reasonable in many applications though it can be problematic in large-scale implementation processing millions of URLs per day [28]. The size of training set linearly grows with memory requirement of pure lazy learners, unlike eager models where the training data are coded into fixed-parameter models. In the case of the dataset under evaluation that contains 1,781 cases and 18 features, storage overhead is not excessive at 15.6 megabytes when using KNN. Nevertheless, production systems that have been running across long historical data might need case base pruning schemes or production architectures that improve both eager and lazy elements [24].

These findings when compared to the previous ensemble learning studies [25] indicate intriguing tradeoffs. The best result of XGBoost was 98.31% accuracy which is around 0.8 percentage points higher than 97.47 of KNN in the past. Nevertheless, KNN has negligible training time and has better interpretability with reference to similar past examples. This tradeoff indicates that the best choice of algorithm would be based on the priorities of the operations. Environments with greater focus on maximum detection accuracy would be well served by ensemble boosting methods, whereas situations where quick model updating, transparent decision-making or limited resources deployment are required would be better served by lazy learning methods [28], [30]. The small standard deviation of the KNN cross-validation scores implies that they do not vary over the various partitions of a dataset, and the observed variation in performance is not due to data artifacts or overfitting.

A number of restrictions deserve to be mentioned. The analysis uses one dataset and has time restrictions of data that had been gathered between 2019 and 2020, which might not be generalizational to the current threats landscapes [30]. Findings would be reinforced by cross-dataset validation based on more current threat intelligence. The study is entirely about feature based classification, which does not involve deep semantic meaning analysis of web page content, or even analysis of JavaScript code [21], [22]. The future research opportunities encompass adopting hybrid architectures based on lazy and eager learning to employ the complementary benefits [24], resiliency to adversarially-engineered URLs aimed at exploiting distance metrics [31], incremental learning variants that use continuously growing threat intelligence [14], explicit case referencing as part of CBR to achieve better interpretability [29] and approximate nearest-neighbor algorithms that can be deployed at enterprise scale [28].

6. CONCLUSION

This study compares lazy learning algorithms to detect malicious URLs in great detail, which fills a significant gap in cybersecurity literature. The study is rigorously evaluated, with systematic experimental design through using unified preprocessing protocols and extensive evaluation metrics, to motivate K-Nearest Neighbors, Locally Weighted Learning, and Case-Based Reasoning methods. Results show that KNN has a higher performance of 97.47 accuracy and 96.92 precision with a recall of 98.15 and is better than the LWL and CBR options besides being well-computationally efficient with little training overhead of 0.47 seconds and prediction latency of 2.83 milliseconds per sample. The inherent interpretability of the algorithm with clear mention of historical cases makes it a feasible choice when it comes to situations which demand model updating speed and clear-cut decision making.

Relevance of features analysis proves the presence of discriminative power of lexical features, especially length of URLs and distributions of special characters, which justifies feature engineering strategies used in previous studies. Although the accuracy of eager learning ensemble methods is marginally better, lazy learning paradigms have other known benefits such as the ability to incorporate new threats information directly, less training computational cost, and the ability to make explicit references to historical cases enabling security analyst interpretation. The study adds empirical results on the instance-based classification models, sets the performance standards to achieve in the future research, and finds the potential areas of development of the hybrid architecture, adversarial robustness testing, and optimization of scalability. The analysis of ROC-AUC shows that it has great discrimination abilities with values greater than 0.96 in all the tested algorithms, and the cross-validation shows that it is statistically reliable with small confidence interval variations that demonstrate similarity in its performance under various data partitions. Since the evolving nature of the cyber threats remains in their advanced and diversified nature, the adaptive nature of lazy learning methods deserves further research and development to achieve holistic cybersecurity systems.

7. REFERENCES

[1] A. Kharraz, W. Robertson, and E. Kirda, "Surveying the landscape of web-based cryptocurrency mining," in Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Toronto, Canada, 2018, pp. 1-15.

[2] S. Yadav, A. K. K. Reddy, A. L. Reddy, and S. Ranjan, "Detecting algorithmically generated malicious domain names," in Proceedings of the ACM SIGCOMM Internet Measurement Conference, Melbourne, Australia, 2010, pp. 48-61.

[3] R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," Neural Computing and Applications, vol. 31, no. 8, pp. 3851-3873, August 2019.

[4] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "PhishNet: Predictive blacklisting to detect phishing attacks," in Proceedings of the IEEE INFOCOM, San Diego, CA, USA, 2010, pp. 1-5.

[5] M. Khonji, A. Jones, and Y. Iraqi, "A study of feature subset evaluators and feature subset searching methods for phishing classification," in Proceedings of the 8th International Conference on Innovations in Information Technology, Al Ain, UAE, 2011, pp. 135-140.

[6] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: A content-based approach to detecting phishing web sites," in Proceedings of the 16th International Conference on World Wide Web, Banff, Canada, 2007, pp. 639-648.

[7] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: A literature survey," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2091-2121, Fourth Quarter 2013.

[8] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious URL detection using machine learning: A survey," arXiv preprint arXiv:1701.07179, 2017.

[9] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," in Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 2009, pp. 1245-1254.

[10] A. Le, A. Markopoulou, and M. Faloutsos,

"PhishDef: URL names say it all," in Proceedings of the IEEE INFOCOM, Shanghai, China, 2011, pp. 191-195.

[11] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment," Computer Communications, vol. 175, pp. 47-57, November 2021.

[12] T. G. Dietterich, "Ensemble methods in machine learning," in Proceedings of the International Workshop on Multiple Classifier Systems, Cagliari, Italy, 2000, pp. 1-15.

[13] Z. H. Zhou, Ensemble Methods: Foundations and Algorithms. Boca Raton, FL: CRC Press, 2012.

[14] T. Mahmood and T. S. Afzal, "Security analytics: Big data analytics for cybersecurity - A review of trends, techniques and tools," in Proceedings of the 2nd National Conference on Information Assurance, Rawalpindi, Pakistan, 2013, pp. 129-134.

[15] A. Al Tamimi, "Detecting phishing URLs using machine learning techniques," International Journal of Computer Science & Network Security, vol. 22, no. 6, pp. 374-380, June 2022.

[16] R. S. Rao, T. Vaishnavi, and A. R. Pais, "CatchPhish: Detection of phishing websites by inspecting URLs," Journal of Ambient Intelligence and Humanized Computing, vol. 11, pp. 813-825, 2020.

[17] W. Ali and S. Malebary, "Particle swarm optimization-based feature weighting for improving intelligent phishing website detection," IEEE Access, vol. 8, pp. 116766-116780, 2020.

[18] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Identifying suspicious URLs: An application of large-scale online learning," in Proceedings of the 26th International Conference on Machine Learning, Montreal, Canada, 2009, pp. 681-688.

[19] D. Sahoo, C. Liu, and S. C. H. Hoi, "Feature-based phishing websites detection using machine learning," Annals of Data Science, vol. 6, no. 1, pp. 145-169, March 2019.

[20] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5-32, October 2001.

[21] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. González, "Classifying phishing URLs using recurrent neural networks," in Proceedings of the APWG Symposium on Electronic Crime Research, Scottsdale, AZ, USA, 2017, pp. 1-8.

[22] W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, and M. Woźniak, "Accurate and fast URL phishing detector: A convolutional neural network approach," Computer Networks, vol. 178, article 107275, August 2020.

[23] R. Vinayakumar, K. P. Soman, P. Poornachandran, and S. Sachin Kumar, "Evaluating deep learning approaches to characterize and classify malicious URL's," Journal of Intelligent & Fuzzy Systems, vol. 34, no. 3, pp. 1333-1343, 2018.

[24] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning," Artificial Intelligence Review, vol. 11, no. 1-5, pp. 11-73, February 1997.

[25] L. Zhang, H. Wang, M. Li, and X. Chen, "Hybrid ensemble learning with deep feature extraction for advanced malware detection," IEEE Transactions on Information Forensics and Security, vol. 19, pp. 3847-3862, 2024.

[26] A. Aggarwal, "Learning to use operational memory for solving binary classification problems," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 3, no. 2, pp. 143-153, April 2019.

[27] K. Bache and M. Lichman, "UCI Machine Learning Repository," University of California, Irvine, School of Information and Computer Sciences, 2013. [Online

[28] H. Zhang and J. Wang, "Scalable k-NN graph construction for fast approximate nearest neighbor search," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 8, pp. 5234-5249, August 2024.

[29] R. López de Mantaras, D. McSherry, D. Bridge, D. Leake, B. Smyth, S. Craw, B. Faltings, M. L. Maher, M. T. Cox, K. Forbus, M. Keane, A. Aamodt, and I. Watson, "Retrieval, reuse, revision and retention in case-based reasoning," The Knowledge Engineering Review, vol. 20, no. 3, pp. 215-240, September 2005.

[30] N. Tariq, M. Asim, F. Al-Obeidat, M. Z. Farooqi, T. Baker, M. Hammoudeh, and I. Ghafir, "The security of big data in fog-enabled IoT applications including blockchain: A survey," Sensors, vol. 19, no. 8, article 1788, April 2019.

[31] M. S. Alam, S. T. Vuong, and R. Pham, "Adversarial attacks against URL-based classifiers: Challenges and defenses," in Proceedings of the IEEE International Conference on Communications, Denver, CO, USA, 2024, pp. 1-6.