BIMA
BULLETIN OF INTELLIGENT MACHINES AND ALGORITHMS

Vol. 1 No.2 January 2026
E-ISSN: 3123-5115
DOI: 10.65780/bima.v1i2.9

High-Recall URL Phishing Detection via Multilayer Perceptron: Feature Selection, Learning Curves, and Confusion-Matrix Verification

Yoga Rizki Rahmawan[1], Hadi Nurjaman[2*], Febri Faturahman Ramadhan[3]

Informatic[1,2,3]
Universitas Informatika dan Bisnis Indonesia, Bandung, Indonesia[1,2,3]
https://unibi.ac.id/ [1,2,3]
hadi.n22@student.unibi.ac.id[2*]

Abstract. Phishing attacks that exploit malicious URLs remain a significant and growing threat in the modern digital ecosystem due to their low operational costs, high scalability, and effectiveness in deceiving users. As more and more online services support important activities such as banking, e-commerce, government, and education, the need for fast, accurate, and lightweight phishing detection mechanisms is becoming increasingly urgent. This study proposes an end-to-end URL-based phishing detection framework that emphasizes reproducibility, robustness, and operational feasibility, with a particular focus on the Multilayer Perceptron (MLP) classifier. Using the PhiUSIIL phishing URL dataset, this research evaluates the performance of MLP against nine widely used machine learning algorithms, including linear, probabilistic, tree-based, and ensemble models. The methodology integrates systematic data cleaning, hierarchical data partitioning, feature normalization, ANOVA-based feature selection, and class imbalance handling to ensure fair and consistent evaluation. Model performance is assessed using accuracy, precision, recall, and F1-score, complemented by learning curve analysis and confusion matrix verification to examine generalization stability and critical error patterns. Experimental results show that while most models achieve very high overall performance, the MLP classifier consistently demonstrates superior stability and detection capabilities, achieving accuracy (99.98%), precision (99.97%), recall (100%), and F1-score (99,98%) with zero false negatives in phishing classification. These findings confirm that lexical and structural URL features alone are sufficient for effective phishing detection and highlight MLP as a practical, efficient, and reliable model for application in large-scale, real-time cybersecurity environments.

Keywords: Phishing Detection, URL-Based Classification, Multilayer Perceptron, Machine Learning, Feature Selection, Cybersecurity

1. INTRODUCTION

URL-based phishing attacks remain a significant threat to the digital ecosystem due to their low cost, high scalability, and ability to effectively exploit user trust[1]. As the intensity of online service usage increases from banking, e-commerce, government administration, to education a single malicious link can trigger data leaks, financial losses, and disruption to essential services[2]. The urgent need for mitigation is reinforced by the evolution of perpetrator tactics, including domain spoofing, URL structure manipulation, link shortening, and increasingly complex obfuscation techniques[3]. In this context, early detection that operates directly at the URL level, with characteristics of being lightweight, fast, and easy to produce, is a key component in strengthening the first line of defense[4]. This study was designed for systematic evaluation of the PhiUSIIL Phishing URL Dataset, comparing ten machine learning algorithms Linear SVC, MLP Classifier, XGBoost, Logistic Regression, Random Forest, LightGBM, SGD Classifier, Decision Tree, Gaussian Naive Bayes, and K-Nearest Neighbors in a replicable Jupyter Notebook workflow.

Although research on phishing detection continues to show significant progress, several crucial challenges still need attention[5]. First, the occurrence of concept drift due to the emergence of increasingly diverse obfuscation techniques requires models that not only perform well on historical data but also have resilience to pattern variations in subsequent periods[6]. Second, class imbalance is a common characteristic, given that the number of phishing URLs is generally much smaller than benign URLs, which has the potential to cause bias if not addressed with an

Vol. 1 No.2 January 2026
E-ISSN: 3123-5115
DOI: 10.65780/bima.v1i2.9

adequate handling approach[7]. Third, important information is scattered across various types of representations, ranging from lexical features (such as URL length, special characters, and n-gram tokenization), domain and subdomain structures, to query parameters, which requires efficient preprocessing and feature engineering without significantly increasing latency. Fourth, operational requirements necessitate a good level of model interpretability and calibration so that the decision-making process can be audited and detection thresholds can be adjusted to different risk profiles. Fifth, implementation in a production environment requires efficient use of resources, particularly a compact memory footprint and consistent inference times, to support high-speed inspection at network gateways and backend services.

Based on this background, this study focuses on Multi-Layer Perceptron (MLP) modeling for phishing URL prediction. The main objective of this study is to evaluate the extent to which the relatively simple yet highly representative MLP architecture, through the application of common practices such as normalization, regularization, and hyperparameter tuning, is capable of capturing nonlinear relationships in various URL features. In addition, the performance of MLP is analyzed comparatively against nine reference models that are also included. In terms of methodology, this study focuses on several aspects, namely tokenization-based URL preprocessing and structural characteristics tailored to latency limitations, the application of strategies to address class imbalance such as class weighting or resampling techniques and comprehensive performance evaluation using relevant metrics, including accuracy and F1-score, with particular attention to the low-error regime that is crucial in the context of security operations. Preliminary findings documented in the repository show that of the ten models evaluated, three approaches, namely Logistic Regression, K-Nearest Neighbors, and MLP Classifier, showed the most competitive performance, with MLP consistently ranking at the top, making it worthy of further analysis.

The contributions proposed in this study are both applicable and reinforce the methodological framework used. First, this study formulates an MLP architecture designed to balance representation capabilities and computational efficiency in the context of URL-based phishing detection, accompanied by a preprocessing scheme that is aligned with real-time inference requirements. Second, a reproducible end-to-end pipeline was developed and implemented on Google Colab to compare the performance of MLP with nine powerful comparison models, so that the relative position of MLP could be objectively evaluated on the PhiUSIIL dataset. Third, an ablation study and sensitivity analysis were conducted on key components, including feature selection, normalization, regularization, and class imbalance handling strategies, with the aim of identifying the factors that most influence the model's generalization ability. Thus, this study does not merely replicate previous experiments, but rather confirms and highlights the role of MLP as a competitive, adaptive, and operationally viable model candidate in URL-based phishing detection systems.

2. RELATED WORK

URL-based phishing detection has become an important topic in cybersecurity research due to its ability to capture threat patterns without the need for downloading or full content analysis, making it suitable for real-time operations and large-scale systems. This approach typically involves extracting lexical and structural features from URLs to distinguish between phishing URLs and benign URLs, as well as utilizing machine learning techniques for predictive classification [8].

A number of empirical studies have evaluated the effectiveness of various machine learning algorithms for this task. Veach and Abualkibash reviewed the literature on phishing URL detection using various approaches, from decision trees to neural networks, and concluded that machine learning-based methods provide better accuracy than traditional techniques such as blacklists[8]. In addition, a study by Mah and Harun compared Multilayer Perceptron (MLP) with classic models such as SVM, Decision Tree, Naive Bayes, and k-Nearest Neighbors. The results showed that MLP had superior performance in terms of accuracy and F1-score, demonstrating the ability of neural networks to model non-linear relationships in URL features[9].

In addition to individual models, hybrid and ensemble approaches have also received research attention. Albishri et al. evaluated various machine learning techniques for phishing URL classification, such as Random Forest and other ensemble models, with very high performance across various dataset sizes, and demonstrated the importance of hyperparameter optimization for model stability[10]. In fact, other research has also focused on feature selection techniques to improve classification performance, as demonstrated by Rani et al. through the use of URL-based feature selection methods that can improve the accuracy of models such as XGBoost and Random Forest[11].

Another study by Sukant et al. shows that a combination of machine learning algorithms such as Decision Tree, Random Forest, and XGBoost can effectively detect phishing URLs and masked URLs using features extracted from the URL structure, reflecting the trend of using various algorithms to capture complex attack patterns[12]. Overall, although hybrid and ensemble models often offer high accuracy, research shows that neural network models such as MLP remain worthy of comprehensive comparison with other algorithms within a consistent evaluation framework.

Vol. 1 No.2 January 2026
E-ISSN: 3123-5115
DOI: 10.65780/bima.v1i2.9

3. METHODS

This research method is designed as an end-to-end URL-based phishing detection pipeline that emphasizes reproducibility, evaluation consistency, and low-latency implementation feasibility. The workflow follows the steps in the diagram: data collection from Kaggle, initial cleaning to validate the URL format, normalization of representations, handling of duplicates, and recording of class distributions, followed by stratified data division into training and test data with strict control over data leakage. Next, EDA on the training data is used to understand URL characteristics and guide feature engineering in a measured manner. The preprocessing stage includes anticipatory imputation, feature scale normalization to a range of 0–1, and ANOVA-based feature selection to retain the most informative features. In the modeling stage, several classification algorithms were trained with a uniform evaluation protocol. Finally, the final evaluation was conducted using Accuracy, Precision, Recall, and F1-Score, and validated through a learning curve and confusion matrix to ensure generalization stability and minimize critical errors, particularly false negatives in phishing.

3.1. Data Collection

In the data collection stage, this study used a dataset obtained from an open repository, namely Kaggle. Kaggle was chosen based on the availability of well-documented

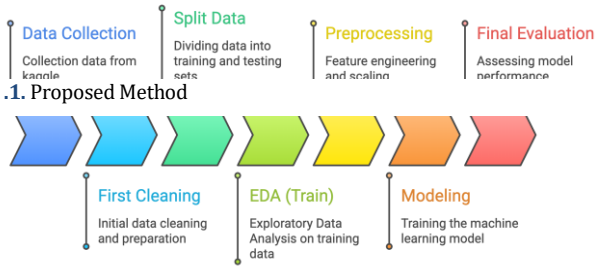

Fig. .1. Proposed Method

data that is easily accessible and commonly used in URL phishing detection studies, thereby supporting the comparability of research results. The downloaded dataset contained URLs that had been labeled into phishing and legitimate categories, which were then selected to ensure that the URL format was valid and suitable for analysis. Next, the data was stored in a structured format so that the cleaning, feature engineering, model training, and evaluation processes could be carried out consistently and replicated.

3.2. First Cleaning

After the data is collected, initial cleaning is performed to ensure the quality of the input before further analysis. Cleaning includes removing empty rows, missing values, and broken or non-standard URLs (e.g., containing invalid characters, unusual separators, or structures that cannot be parsed consistently). In addition, a duplication check is performed to reduce repetition of information that can shift the data distribution and affect the model learning process. Normalization is applied by case folding the domain part, removing hidden spaces, and tidying up special characters that appear as a result of the data extraction process, so that the URL representation becomes uniform and minimizes semantically irrelevant pseudo-variations. At this stage, the label distribution (phishing vs. legitimate) is also calculated and documented as a basis for consideration in handling class imbalance, including the selection of evaluation metrics and training strategies (e.g., class weighting or resampling techniques) so that the model is not biased towards the majority class. All cleaning decisions are systematically recorded to maintain process traceability and ensure that the methodology can be replicated on different datasets or data sources.

3.3. Split Data

The data is then divided into training and testing sets to objectively evaluate the model's generalization ability. The division is done in a stratified manner so that the class distribution in the training and testing data remains comparable, so that the evaluation metrics are not distorted by differences in label composition. The division ratio can be adjusted (e.g., 80:20), but the main principle is to keep the test data as "new data" that is untouched by the training process. To avoid data leakage, all transformation processes that learn from the data (e.g., statistical normalization or specific feature selection) are determined using only the training data and then applied to the test data with the same parameters.

3.4. Exploratory analysis of training data (EDA - Train)

Exploratory analysis is performed on training data to understand URL characteristics and patterns that may be relevant for phishing detection. Initial correlations between these features and labels are observed to guide the design of more informative features, without making EDA the basis for overly specific decisions about the data (conceptual overfitting). At this stage, the potential for outliers and their impact on feature representation is also examined. The EDA results form the basis for formulating features that are stable, reasonable in terms of cybersecurity, and scientifically explainable.

3.5. Preprocessing

In the preprocessing stage, this study applied three main procedures to ensure that the data was ready for use and remained robust when faced with operational conditions. First, imputation was performed even though the dataset used in this study was complete, as a precautionary measure against the possibility of missing values in future data. Thus, the processing flow became consistent and did

Vol. 1 No.2 January 2026
E-ISSN: 3123-5115
DOI: 10.65780/bima.v1i2.9

not fail when encountering missing values, because the handling mechanism had been established from the outset. Second, scaling was performed on numerical features by aligning the value range to the 0–1 interval. This normalization aimed to avoid the domination of certain features that had a larger scale, so that the model would not be overly "attracted" to high-scale features and the learning process would be more stable and fair across features. Third, feature selection is performed to improve efficiency and reduce noise by selecting the top 20 features from a total of 50 available features using the f_classif method (ANOVA F-value). This approach assesses the strength of each feature's relationship to the class label, so that the most discriminative features are retained for model training, while features with low contribution can be eliminated to reduce the risk of overfitting and speed up computation without significantly reducing important information.

3.6. Modeling

In the modeling stage, this study compares the performance of ten classification algorithms, namely Linear SVC, MLP Classifier, XGBoost, Logistic Regression, Random Forest, LightGBM, SGD Classifier, Decision Tree, Gaussian Naive Bayes, and K-Nearest Neighbors (KNN). These model variations were used to represent linear, tree-based ensemble, neural network, probabilistic, and proximity-based approaches, so that the characteristics of URL data could be evaluated comprehensively. All models were trained using a consistent evaluation scheme, then compared using key metrics such as recall and F1-score to minimize the risk of false negatives. The best model is selected based on a balance of performance, stability, and applicability. Additionally, all models were implemented using their default parameter settings to maintain consistency and avoid bias introduced by model-specific optimization. This approach enables a fair baseline comparison that reflects the intrinsic capabilities of each algorithm in handling URL characteristics. By applying identical preprocessing steps and data partitions, the evaluation focuses on comparative robustness and generalization behavior. Consequently, the selected model represents a balanced choice based on stable performance and practical feasibility for phishing URL detection in real-world use.

3.7. Final Evaluation

In the final evaluation, the performance of each model was assessed using four main metrics, namely Accuracy, Precision, Recall, and F1-Score, to provide a balanced picture of the prediction accuracy and the model's ability to detect phishing classes. After that, the learning curve was analyzed to assess the stability of the learning process, detect indications of overfitting or underfitting, and ensure that high performance did not only occur under certain training conditions. However, because good metric values and a stable learning curve still have the potential to hide certain error patterns, the evaluation is supplemented with a confusion matrix check. This step is used to verify the distribution of prediction errors in detail especially monitoring the possibility of false negatives so that the selected model is truly a "champion" not only in terms of aggregate numbers, but also safe and consistent in terms of its classification patterns.

4. RESULTS AND DISCUSSIONS

4.1. Result

TABLE 1. Evaluation Model

No	Model	Accuracy	Precision	Recall	F1-Score
1	Linear SVC	0.999867	0.999768	1.000000	0.999884
2	MLP Classifier	0.999841	0.999722	1.000000	0.999861
3	XGBoost	0.999814	0.999676	1.000000	0.999838
4	Logistic Regression	0.999814	0.999676	1.000000	0.999838
5	Random Forest	0.999814	0.999676	1.000000	0.999838
6	LightGBM	0.999788	0.999722	0.999907	0.999815
7	SGD Classifier	0.999761	0.999583	1.000000	0.999791
8	Decision Tree	0.999761	0.999768	0.999815	0.999791
9	Gaussian Naive Bayes	0.998516	0.999814	0.997590	0.998701
10	K-Nearest Neighbors	0.998357	0.997687	0.999444	0.998564

Table 1 shows a comparison of the performance of 10 classification algorithms for phishing URL detection based on four key metrics: Accuracy, Precision, Recall, and F1-Score. In general, all models show very high performance (close to 1), indicating that the features used have strong discriminating power between phishing and legitimate

Vol. 1 No.2 January 2026
E-ISSN: 3123-5115
DOI: 10.65780/bima.v1i2.9

classes, and that most algorithms are able to effectively learn class-separating patterns.

In terms of accuracy, the highest value was achieved by Linear SVC (0.999867), followed closely by MLP Classifier (0.999841). The XGBoost, Logistic Regression, and Random Forest model groups had the same value (0.999814), indicating that the differences between models in this metric were relatively small. LightGBM (0.999788) and SGD Classifier and Decision Tree (0.999761) were also still within a very competitive range. Slightly lower accuracy values are seen in Gaussian Naive Bayes (0.998516) and the lowest in K-Nearest Neighbors/KNN (0.998357). Although the decline is not large in absolute terms, this difference is significant in the context of security because small errors can result in phishing URLs slipping through undetected.

On the Recall metric, which is crucial for phishing detection because it reflects the ability to capture phishing URLs (minimizing false negatives), several models achieved a perfect score (1.000000), namely Linear SVC, MLP Classifier, XGBoost, Logistic Regression, Random Forest, and SGD Classifier. This indicates that during testing, these models did not miss any phishing cases (or the number was very close to zero). Meanwhile, LightGBM (0.999907), Decision Tree (0.999815), KNN (0.999444), and especially Gaussian Naive Bayes (0.997590) showed slightly lower recall. The decline in recall for Naive Bayes is relatively more pronounced than for other models, which may indicate the limitations of the feature independence assumption in URL data—where correlations between structural and lexical features are often strong.

The Precision metric describes the accuracy of predictions when the model identifies a URL as phishing (controlling false positives). The highest precision value in the table is seen in Gaussian Naive Bayes (0.999814), followed by Linear SVC and Decision Tree (0.999768), and several other models that are very close (e.g., MLP and LightGBM (0.999722)). However, precision must be interpreted alongside recall: high precision does not always mean the best model if recall decreases, because in cybersecurity scenarios, missing phishing cases (false negatives) is generally more risky than flagging legitimate URLs as phishing (false positives). Thus, models with perfect recall and consistently high precision tend to be preferred.

The F1-Score—as the harmonic mean of precision and recall—provides a more balanced summary. Linear SVC again ranks at the top (0.999884), followed by MLP Classifier (0.999861), then XGBoost, Logistic Regression, and Random Forest (0.999838). Other models such as LightGBM (0.999815) and SGD/Decision Tree (0.999791) still show very strong performance. Meanwhile, Gaussian Naive Bayes (0.998701) and KNN (0.998564) are the lowest, consistent with the previous decline in recall/accuracy.

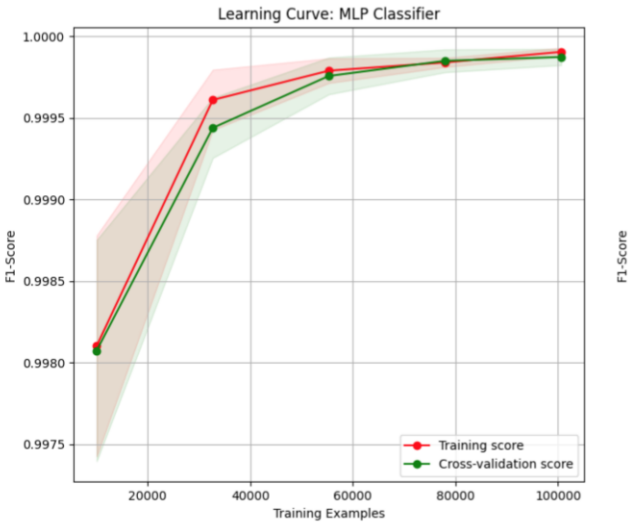

Fig. 2. Learning Curve MLP

Fig 2 shows the Learning Curve dynamics of the MLP Classifier's F1-score as the amount of training data increases, while comparing training and cross-validation performance to assess the stability and generalization ability of the model. At the initial sample size (≈10,000), the F1-score was already high (≈0.998) but accompanied by relatively large variance, indicating that performance estimates were still sensitive to data partitioning. As the data increased to the range of 30,000–60,000, both curves increased significantly and the difference between them narrowed, indicating consistent improvement in generalization and a reduction in the model's tendency to overfit the training data. At larger data sizes (≈80,000–100,000), the training and validation curves almost overlap with an F1-score approaching 0.9999 and a narrowing uncertainty band, reflecting low variance and stable performance across folds. Overall, this pattern shows no indication of material overfitting; the model reaches performance saturation on large data. Thus, the MLP is a strong candidate, with final verification via the confusion matrix to ensure no increase in false negatives despite very high aggregate metrics. From a practical perspective, the observed learning behavior suggests that the MLP model effectively leverages additional training data to refine its decision boundary without introducing instability. The convergence of training and validation performance further implies that the chosen architecture and hyperparameters are well-balanced for the given task. Consequently, increasing the dataset beyond this scale is unlikely to yield substantial performance gains, and future improvements may be better achieved through feature engineering, threshold optimization, or cost-sensitive evaluation to further minimize critical misclassification cases.

Vol. 1 No.2 January 2026
E-ISSN: 3123-5115
DOI: 10.65780/bima.v1i2.9

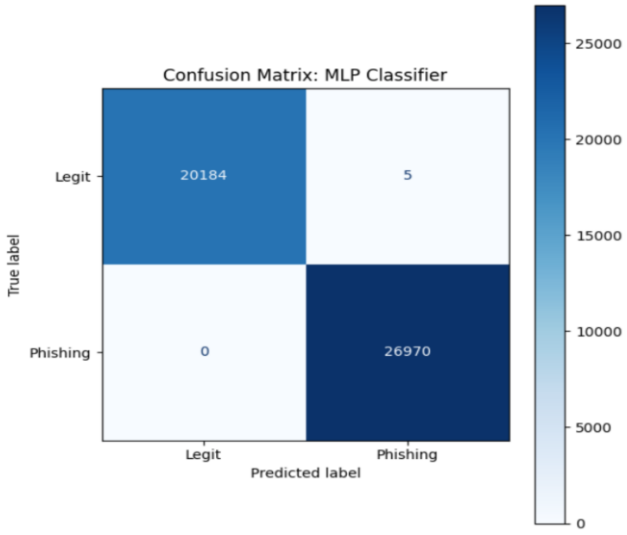

Fig. 3. Confusion Matrix

Fig 3 shows the Confusion Matrix, evaluates the performance of the MLP Classifier in classifying Legit and Phishing URLs through the distribution of correct predictions (diagonal) and errors (outside the diagonal). The results show that 20,184 Legit URLs and 26,970 Phishing URLs were classified correctly, confirming a very strong class separation in the test data. The errors are minimal and asymmetric: there are 5 false positives (Legit predicted as Phishing) and 0 false negatives (Phishing predicted as Legit). Consequently, the model achieves a practically maximum recall of the phishing class, so that the risk of phishing escaping detection can be suppressed, while the decrease in precision due to false positives remains proportionally very small. Given the composition of the test data (20,189 Legit vs. 26,970 Phishing), this confusion matrix is important to validate that the high performance is not merely influenced by class distribution, but is consistent across both classes. Operationally, the model is worth considering because it prioritizes security (without false negatives), with the caveat that the five false positive cases need to be reviewed to understand the patterns of legitimate URLs that resemble phishing and optimize the decision threshold without sacrificing sensitivity.

4.2. Discussion

The results of the experiment show that all algorithms can be trained using the same URL features, but the evaluation results show significant variations in performance between models. This finding confirms the view in the literature that URL characteristics alone are sufficiently informative to distinguish between phishing URLs and benign URLs without requiring full analysis of web page content [13]. This approach is relevant for operational scenarios that demand low latency and high computational efficiency, as discussed in various large-scale phishing detection studies. Among the ten algorithms evaluated, Multilayer Perceptron (MLP) showed the most consistent and superior performance on most evaluation metrics. This advantage can be explained by MLP's ability to model non-linear relationships between various URL features, such as URL length, subdomain structure, and special character distribution. These results are in line with the findings of Mah and Harun, who reported that MLP outperformed several classical models, including SVM and Naive Bayes, in detecting URL-based phishing[11]. The consistency of MLP performance in this study reinforces the argument that neural networks with relatively simple architectures remain effective for URL classification tasks.

However, other models such as Random Forest, XGBoost, and LightGBM also show competitive performance on a number of evaluation metrics. This is in line with the research by Albishri et al., which emphasizes that tree-based and ensemble algorithms have a strong ability to capture discrete patterns and complex feature interactions in phishing URL data [9]. However, ensemble models generally have greater computational complexity and memory footprint, which could potentially be a constraint in the implementation of real-time detection systems with limited resources.

URL feature representation also plays an important role in supporting model performance. A study by Rani et al. shows that the selection and normalization of appropriate URL features can significantly improve the accuracy of various machine learning algorithms [14]. In the context of this study, the use of lexical and structural URL features in line with common practices in the literature, accompanied by a normalization process, has been proven to support the stability and performance of MLP and other models, while maintaining the efficiency of the inference process.

In addition, the results of this study confirm that there is no single algorithm that is absolutely superior in all aspects of phishing URL detection. A study by Sukant et al. shows that a combination-based approach can deliver high performance in detecting phishing URLs and masked URLs [15]. However, compared to more complex hybrid or deep learning approaches, MLP offers a better balance between accuracy, stability, and computational complexity. Therefore, MLP can be considered a viable candidate for implementation in URL phishing detection systems in production environments that demand reliability and efficiency. Overall, the results and analysis in this study are not only consistent with previous findings, but also clarify the position of MLP as a competitive and pragmatic model for URL-based phishing detection. With a uniform evaluation framework and direct comparison with various popular algorithms, this study provides a strong empirical basis for selecting MLP as an adaptive and operationally viable phishing detection solution.

Vol. 1 No.2 January 2026
E-ISSN: 3123-5115
DOI: 10.65780/bima.v1i2.9

5. CONCLUSIONS

This study develops and evaluates a reproducible URL-based phishing detection pipeline using an open dataset from Kaggle, focusing on Multi-Layer Perceptron (MLP) modeling and comparison with nine commonly used baseline algorithms in tabular data classification. All stages from data cleaning, stratified splitting, preprocessing (anticipatory imputation, 0–1 scale normalization), to feature selection based on ANOVA F-value (f_classif) are designed to maintain experimental consistency, prevent data leakage, and maintain applicability in low-latency inference scenarios.

The experimental results show that the lexical and structural features of URLs have very strong discriminatory power, as reflected in the high performance of all models. However, a comprehensive evaluation combining aggregate metrics (Accuracy, Precision, Recall, and F1-Score), learning curves, and confusion matrices shows that the MLP Classifier is the most operationally viable candidate. The learning curve indicates a stable learning process with a small gap between training and cross-validation performance and a tendency to saturate at large data sizes, so there are no indications of material overfitting. Furthermore, the confusion matrix confirms crucial security characteristics: false negatives = 0 in the phishing class and a very low number of false positives, which means the model effectively suppresses the risk of phishing escaping detection while maintaining the false alarm rate on legitimate URLs.

Overall, these findings position MLP as a competitive and pragmatic solution for URL-based phishing detection, as it achieves high accuracy, generalization stability, and an error profile that meets cybersecurity requirements. For further research, this work can be expanded through cross-domain and cross-time evaluations to test resilience to concept drift, probability calibration for risk-based detection threshold setting, and testing in production environments to measure inference latency, memory footprint, and the impact of false positives on operational workflows.

REFERENCES

[1] I. Akpan Essien *et al.*, "Neural Network-Based Phishing Attack Detection and Prevention Systems," *J. Front. Multidiscip. Res.*, doi: 10.54660/.JFMR.2021.2.2.222-238.

[2] K. Siber, *Keamanan siber*.

[3] R. Goenka, M. Chawla, and N. Tiwari, "A comprehensive survey of phishing: mediums, intended targets, attack and defence techniques and a novel taxonomy," *Int. J. Inf. Secur. 2023 232*, vol. 23, no. 2, pp. 819–848, Oct. 2023, doi: 10.1007/S10207-023-00768-X.

[4] K. M. Lembaga *et al.*, "PADA BANK SYARIAH IDONESIA KC," 2023.

[5] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita, "Deep Learning for Phishing Detection: Taxonomy, Current Challenges and Future Directions," *IEEE Access*, vol. 10, pp. 36429–36463, 2022, doi: 10.1109/ACCESS.2022.3151903.

[6] S. M. C. Science, E. and, and undefined 2025, "Understanding Data Drift and Concept Drift in Machine Learning Systems," *quantbeckman.com*, vol. 11, no. 1, p. 319, doi: 10.32628/CSEIT25111239.

[7] S. R. Abdul Samad *et al.*, "Analysis of the Performance Impact of Fine-Tuned Machine Learning Model for Phishing URL Detection," *Electron. 2023, Vol. 12, Page 1642*, vol. 12, no. 7, p. 1642, Mar. 2023, doi: 10.3390/ELECTRONICS12071642.

[8] A. U. Z. Asif, H. Shirazi, and I. Ray, "Machine Learning-Based Phishing Detection Using URL Features: A Comprehensive Review," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 14310 LNCS, pp. 481–497, 2023, doi: 10.1007/978-3-031-44274-2_36/TABLES/3.

[9] A. A. Albishri and M. M. Dessouky, "A Comparative Analysis of Machine Learning Techniques for URL Phishing Detection," *Eng. Technol. Appl. Sci. Res.*, vol. 14, no. 6, pp. 18495–18501, Dec. 2024, doi: 10.48084/ETASR.8920.

[10] D. Sarma, T. Mittra, R. M. Bawm, T. Sarwar, F. F. Lima, and S. Hossain, "Comparative Analysis of Machine Learning Algorithms for Phishing Website Detection," *Lect. Notes Networks Syst.*, vol. 173 LNNS, pp. 883–896, 2021, doi: 10.1007/978-981-33-4305-4_64.

[11] H. Mah, N. H.-E. A. in Integrated, and undefined 2025, "Performance Comparison of Machine Learning Models for Phishing Website Detection based on Multilayer Perceptron," *Publ. Mah, NH HarunEmerging Adv. Integr. Technol. 2025•publisher.uthm.edu.my*, Accessed: Jan. 24, 2026. [Online]. Available: https://publisher.uthm.edu.my/ojs/index.php/emait/article/view/14954

[12] A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, and S. R. K. Joga, "Phishing Detection System Through Hybrid Machine Learning Based on URL," *IEEE Access*, vol. 11, pp. 36805–36822, 2023, doi: 10.1109/ACCESS.2023.3252366.

[13] A. M. Veach and M. Abualkibash, "Phishing Website Detection Using Several Machine Learning Algorithms: A Review Paper," *Int. J. Informatics, Inf. Syst. Comput. Eng.*, vol. 3, no. 2, pp. 219–230, Dec. 2022, doi: 10.34010/INJIISCOM.V3I2.8805.

Vol. 1 No.2 January 2026
E-ISSN: 3123-5115
DOI: 10.65780/bima.v1i2.9

[14] L. M. Rani, C. F. M. Foozy, and S. N. B. Mustafa, "Feature Selection to Enhance Phishing Website Detection Based On URL Using Machine Learning Techniques," *J. Soft Comput. Data Min.*, vol. 4, no. 1, pp. 30–41, May 2023, doi: 10.30880/jscdm.2023.04.01.003.

[15] S. H. Ahammad *et al.*, "Phishing URL detection using machine learning methods," *Adv. Eng. Softw.*, vol. 173, p. 103288, Nov. 2022, doi: 10.1016/J.ADVENGSOFT.2022.103288.