iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.71858

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

A Novel Machine Learning Approach for Malicious URL Classification

P. Sukumar Reddy[1], K. Balakrishna Maruthiram[2]
[1]*Post Graduate Student, M.Tech(CNIS),Department of Information Technology, Jawaharlal Nehru Technological University, Hyderabad, India*
[2]*Assistant Professor, Department of Information Technology, Jawaharlal Nehru Technological University, Hyderabad,India*

Abstract: ***Exponential rise in cyber threats has made malicious URL detection an essential component of modern cybersecurity systems. Malicious URLs are frequently used to initiate phishing attacks, spread malware, or deface legitimate websites, posing severe risks to users and organizations. In this research, we propose a hybrid machine learning approach for the efficient classification of URLs into four categories: benign, phishing, malware, and defacement. The system leverages both traditional machine learning algorithms and advanced deep learning techniques. Lexical features such as URL length, the number of special characters, and HTTPS presence are extracted and used to train classifiers including Decision Tree, Random Forest, and XGBoost. In parallel, a Character-level Convolutional Neural Network (Char-CNN) is developed to process raw URL strings, enabling automatic feature extraction at the character level.A labeled dataset containing over 599,359real-world URLs is used for training and evaluation. The models are assessed using standard performance metrics such as accuracy, precision, recall, and F1-score. Experimental results show that the Char-CNN model achieves superior accuracy compared to traditional models, while Random Forest and XGBoost demonstrate robust and interpretable performance. The combination of feature-based and character-level models ensures both high detection accuracy and adaptability to evolving attack patterns. The proposed system can be effectively integrated into real-time web security applications to detect and block malicious URLs with high reliability.***
Keywords: Malicious URL Detection, Phishing, Machine Learning, Random Forest, XGBoost, Character-level CNN, Cybersecurity

I. INTRODUCTION

In the current digital era, the internet has become an essential part of daily life, enabling communication, commerce, education, and information sharing. However, the growing dependence on web services has also led to a significant increase in cyber threats. Among these, malicious URLs have emerged as one of the most prevalent and dangerous vectors used by attackers to carry out phishing attacks**,** malware distribution**,** website defacement, and social engineering exploits**.** These URLs are crafted to deceive users and direct them to harmful content, often without their knowledge. Traditional techniques for detecting malicious URLs, such as blacklisting**,** heuristic rules**,** and signature-based detection, suffer from several limitations. They rely on pre-identified patterns or known threat databases, making them ineffective against zero-day attacks**,** dynamically generated URLs, and obfuscated links**.** Additionally, these methods lack scalability and adaptability, which are essential for defending against the constantly evolving tactics of modern cyber attackers.

To address these limitations, recent advancements in machine learning (ML)and deep learning (DL**)** have provided a more intelligent and automated approach to detecting malicious URLs. Unlike traditional methods, ML and DL techniques can learn patterns and behaviours from large datasets, enabling them to generalize beyond known threats and detect previously unseen malicious URLs.This project presents **a** hybrid system for malicious URL detection that leverages both classical machine learning algorithms and deep learning architectures to achieve high accuracy and robustness. Specifically, the system integrates Decision Tree (DT)**,** Random Forest (RF**)**, and XGBoost for processing lexical features extracted from URLs, and a Character-level Convolutional Neural Network **(**Char-CNN**)** for analyzing raw URL strings without the need for manual feature extraction.

The ML models use engineered features such as URL length**,** number of special characters**,** presence of IP address, use of HTTPS**,** and URL shortening**.** These features are fast to extract and provide a strong signal for classifying URLs as benign**,** phishing**,** malware**,** or defacement. Meanwhile, the Char-CNN model tokenizes each URL at the character level and uses convolutional layers to learn URL patterns, making it especially effective in identifying obfuscated or novel URLs**.**

The combination of ML and DL models in a hybrid framework ensures that the system achieves both speed and accuracy. The machine learning models are computationally efficient and interpretable, making them suitable for real-time applications, while the deep learning model enhances the system's ability to detect complex and previously unseen patterns. To train and evaluate the system, a publicly available dataset from Kaggle was used, containing over 250,000 labeled URLs. The dataset includes four classes of URLs: benign, phishing, malware, and defacement. The models were trained and tested using standard performance metrics such as accuracy**,** precision**,** recall**,** F1-score**,** and confusion matrix, achieving highly promising results. The Char-CNN model, in particular, reached an accuracy of over 99%**,** while Random Forest and XGBoost also demonstrated strong performance.

This project demonstrates the effectiveness of combining structured lexical feature-based ML models with unstructured deep learning models for malicious URL detection. The proposed system can be integrated into real-time security solutions such as web browsers, email filters, or network firewalls to protect users against a wide range of cyber threats.

.

II. LTERATURE REVIEW

With the increasing dependency on the internet, cyber threats have evolved in both volume and sophistication. One of the most common and dangerous threats is the use of malicious URLs**,** which are designed to deceive users and carry out phishing, malware downloads, defacements, or other forms of cyberattacks. Traditional detection methods like blacklists and signature-based approaches are limited due to their inability to detect zero-day attacks**.** Consequently, researchers have turned to machine learning(ML) and deep learning (DL) approaches to build intelligent systems that can detect malicious URLs based on their structural and behavioural patterns.

A. Machine Learning-Based Detection Approaches

Several researchers have explored the use of supervised learning algorithms for malicious URL detection. Ma et al. (2011) were among the first to propose using lexical and host-based features for classification. They tested various classifiers and found that Online Learning methods were both scalable and effective. Similarly, Sahoo et al. (2017) presented a detailed survey outlining how ML techniques outperform traditional heuristics by learning patterns from previously labeled data.

In our current project, we have extended the approach by using algorithms like:

- Decision Tree (DT)
- Random Forest (RF)
- XGBoost

These classifiers have proven effective in previous studies. Random Forest, in particular, is known for its robustness high accuracy and to overfitting (as seen in Patil et al., 2018). XGBoost, a more recent gradient boosting framework, is praised for its computational efficiency and precision, and has been widely adopted in many cybersecurity tasks, including malicious URL detection.

B. Feature Engineering and Lexical Analysis

Most ML-based systems depend on carefully engineered features derived from the URL string. Lexical features like URL length,number of special characters (e.g., **@**, **-**, **?**), presence of IP address, HTTPS usage, and URL shortening are commonly used. These features are easy to extract and don't require querying external servers, making them suitable for real-time systems**.**

Garera et al. (2007) emphasized that phishing URLs often have distinguishing lexical properties and demonstrated that a logistic regression model based on such features could be highly effective. Our system uses similar lexical features, encoded into numerical form, which are then used by the ML models.

C. Deep Learning and Character-Level Modeling

In contrast to ML models, deep learning techniques can operate on raw data without manual feature extraction. Character-level Convolutional Neural Networks (Char-CNN**)** are particularly suited for URL classification as they can learn URL syntax patterns directly from the text.

Vinayakumar et al. (2018) evaluated CNN and LSTM architectures and demonstrated their superior performance in classifying obfuscated or previously unseen URLs. More recently, Srinivasan et al. (2021) developed DURLD, a deep learning framework that uses character-level representations for robust malicious URL detection.

Inspired by these studies, our system incorporates a Character-level CNN with:

- Character-level tokenization
- Embedding layer
- Multiple convolution and pooling layers
- Dense output for multi-class classification

This deep learning component enhances our detection performance, especially for complex or obfuscated URLs.

D. Hybrid Approaches

Few works have attempted to combine both ML and DL approaches. While ML models are fast and interpretable, DL models are powerful and generalize better. Our approach integrates both to balance interpretability, accuracy, and performance. For instance:

- RF/XGBoost are used with lexical features for fast and interpretable predictions.
- Char-CNN complements this with its strength in handling unseen patterns and encoding structure automatically.

This hybrid strategy aligns with modern trends in cybersecurity systems that leverage the best of both classical and neural approaches.

III. PROPOSED WORK

The proposed work introduces a hybrid machine learning-based system for detecting and classifying malicious URLs into four categories: benign, phishing, malware, and defacement. This system combines traditional machine learning algorithms with a deep learning model to leverage both structured feature analysis and character-level URL understanding. The process begins with the preparation of a labeled dataset consisting of over 600,000 URLs, where preprocessing steps such as null value removal and duplicate elimination are applied.

To support the traditional models, lexical features are extracted from each URL, including length, number of special characters (such as dots, slashes, hyphens), presence of HTTPS, and detection of IP addresses. These features are then used to train classifiers like Decision Tree, Random Forest, and XGBoost. Simultaneously, a Character-level Convolutional Neural Network (Char-CNN) is implemented to analyze raw URL strings directly, using character embeddings and convolutional layers to learn patterns automatically without manual feature engineering.Both sets of models are trained and evaluated using standard performance metrics such as accuracy, precision, recall, and F1-score.

Grid Search is used for hyperparameter optimization in traditional models, while the Char-CNN is trained over multiple epochs using a custom batch generator.

The results demonstrate that Char-CNN achieves superior detection accuracy, while Random Forest and XGBoost provide reliable and interpretable performance as baselines. real-time URL prediction function is developed, allowing the system to classify any input URL using either the lexical feature-based models or the deep learning model. This hybrid approach ensures high accuracy, adaptability, and practical usability for real-time malicious URL detection in cybersecurity applications

A. Dataset Description

The dataset employed in this project serves as the foundation for training, validating, and testing machine learning and deep learning models for malicious URL detection. It is a publicly available, real-world dataset consisting of 599,359 URL entries, each labeled to indicate whether it is benign (safe)or malicious—the latter including subtypes such as phishing**,** malware**,** and defacement. Due to its richness and diversity, this dataset is frequently used in academic and industry research related to cybersecurity, particularly in threat intelligence and URL-based classification systems.

Each data entry contains two core components:

- The URL string**,** representing the web address
- The label**,** denoting the threat category associated with the URL

This dataset allows for multi-class classification**,** which is more realistic and practical compared to binary classification (i.e., simply malicious or not). By classifying URLs into specific categories, the detection system not only identifies the presence of a threat but also its nature, enabling better decision-making and threat response in security systems.

Category 1: Class Distribution

To ensure a well-balanced training process, a subset of 250,000 samples was selected from the full dataset. This subset includes:

- Benign URLs: 182,831
- Defacement URLs: 46,218
- Phishing URLs: 14,970
- Malware URLs**:** 5,981

This class distribution introduces some degree of class imbalance, which is common in real-world threat datasets. Appropriate techniques such as stratified sampling, resampling, or class weighting can be used to address this during training.

Category 2:Data Preprocessing

Prior to model training, the dataset underwent essential preprocessing steps to ensure consistency and quality of input:

- For machine learning models, lexical features were extracted and normalized or scaled where necessary to bring them into compatible numeric ranges.
- URLs were tokenized at the character level to preserve structure and patterns.
- Sequences were padded or truncated to a fixed length for uniformity across inputs.
- The character tokens were passed through an embedding layer, enabling the model to learn abstract representations and detect patterns in character sequences.

These preprocessing steps ensured that the dataset was appropriately structured for both feature-based machine learning algorithms and sequence-based deep learning models.

Category 3:Feature Engineering

To prepare the dataset for traditional machine learning models, various lexical features were extracted from each URL. These features aim to capture structural patterns often found in malicious URLs. The following summarizes the key features used:

To effectively detect malicious URLs, a set of handcrafted features was extracted based on the structural and lexical characteristics of URLs. These features help in identifying suspicious patterns commonly found in phishing, malware, and defacement URLs. Below is a list of the features used in this study:

- **url_length**: Total number of characters in the URL.
- **num_dots**: Number of dot (.) characters in the URL.
- **num_slashes**: Number of slash (/) characters in the URL.
- **num_question_marks**: Number of question mark (?) characters, often indicating URL parameters.
- **num_hyphens**: Number of hyphen (-) characters, sometimes used in obfuscated URLs.
- **num_at**: Number of at (@) characters, often used in phishing attempts to hide domains.
- **num_equals**: Number of equal sign (=) characters, usually used in URL query parameters.
- **has_https**: Indicates whether the URL contains https (1 = yes, 0 = no).
- **is_ip**: Indicates if the domain is an IP address (1 = yes, 0 = no).
- **type (target label)**: Encoded class label representing the type of URL

http://www.gst-ukraine.com.ua/ukraine/tours/2754-arboretum-alexandria-sofiyivka	defacement
facebook.unitedcolleges.net	phishing
http://www.bimabn.com/1-configurazione-supporto-apple.store-contatta/c/Apple-id/3d465e25b47e6bc23ae55f5de40e	phishing
eyeappealoptometry.com/	benign
brophygen.com/	benign
icehockey.wikia.com/wiki/Dan_Blackburn	benign
gurufocus.com/news/116068/who-is-the-warren-buffett-of-canada-prem-watsa-of-fairfax-financial-ffh-or-paul-desmara	benign
http://www.bg-ricevimenti.it/index.php/location/osteria-del-molo.html?fontstyle=f-smaller	defacement
stewstew.com/	benign
livingharvest.com/products/ice-cream-bars	benign
http://icicibank.com/Personal-Banking/offers/offer-detail.page?id=offer-readers-digest-magazine-offer-20150202203953	benign
http://9779.info/%E5%84%BF%E7%AB%A5%E7%AB%8B%E4%BD%93%E7%BA%B8%E8%B4%B4%E7%94%BB/	malware

Table-1:DatasetExample

B. Architecture

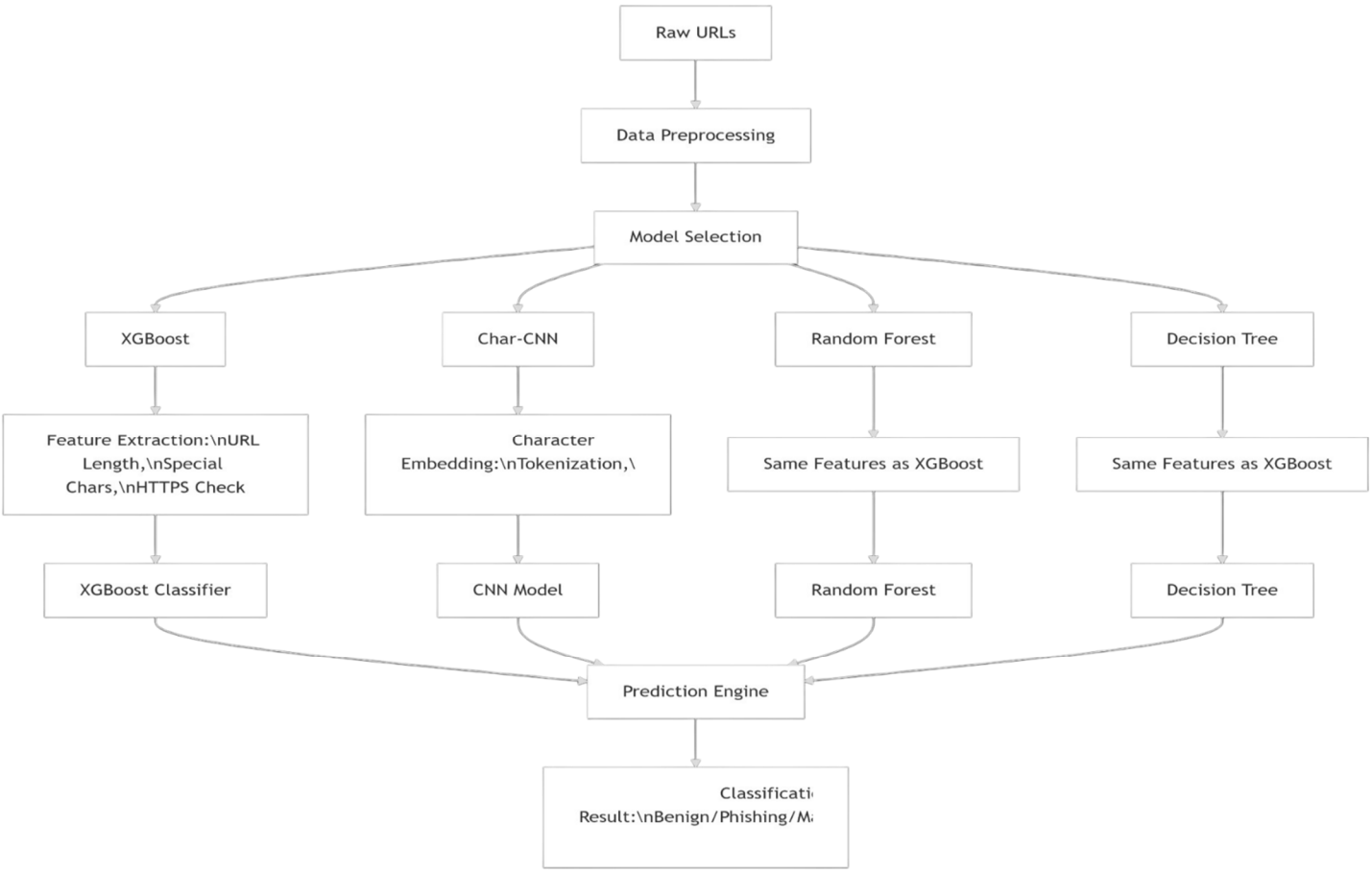


Figure-1: Architecture

The proposed architecture for malicious URL detection is composed of multiple components organized into a modular and parallel processing structure. The workflow is designed to process raw URLs, extract relevant features, apply different classifiers, and ultimately produce a prediction indicating whether a URL is benign, phishing, or malware. The complete architecture is outlined below

1. Raw URLs
   - The input to the system consists of a large dataset of raw URLs collected from diverse sources, which includes both benign and malicious URLs (phishing and malware). These URLs are the foundation for model training and prediction.

2. Data Preprocessing

This stage ensures that raw URLs are cleaned and transformed appropriately for downstream tasks. It involves:

- Normalization: Removal of irrelevant characters and standardization of URL format.
- Tokenization (for Char-CNN): Conversion of URLs into sequences of characters.
- Label encoding: Mapping class labels (Benign, Phishing, Malware) to numeric values.
- Padding (for Char-CNN): Ensures uniform sequence length by padding shorter URLs.

3. Model Selection

The system evaluates multiple models, each following a distinct pathway. There are four main classifiers used in parallel:

- XGBoost
- Character-level Convolutional Neural Network (Char-CNN)
- Random Forest
- Decision Tree

4. Feature Extraction / Embedding

- XGBoost, Random Forest, and Decision Tree utilize lexical features such as URL length, number of special characters, presence of HTTPS, IP address, and suspicious words.
- Char-CNN processes raw URLs through character-level tokenization, embedding layers, and CNN filters to extract local sequential patterns.
- These approaches combine manual feature engineering with deep learning for effective URL classification.

5. Classification Models

Each model uses the extracted features or embeddings to perform classification:

- XGBoost Classifier: A gradient boosting model optimized for performance and accuracy.
- CNN Model (Char-CNN): Learns character-level patterns directly from the input without manual feature engineering.
- Random Forest: An ensemble of decision trees using bagging and feature randomness.
- Decision Tree: A simple, interpretable model using learned rules based on feature thresholds.

6.Prediction Engine

The outputs from all models are forwarded to a centralized Prediction Engine, which:

- Aggregates predictions from all classifiers.
- Applies either majority voting or confidence-based selection to produce a final label.

7.Output Classification

The final output is the classification of the input URL into one of the following categories:

- Benign
- Phishing
- Malware

This result aids in threat identification and informs security responses for further action

This multi-model architecture enhances the robustness and accuracy of URL classification by leveraging both traditional machine learning and deep learning approaches. While lexical feature-based models offer speed and interpretability, the Char-CNN model complements them with its ability to learn complex patterns directly from raw URL strings. The ensemble design ensures better generalization and resilience to evolving threats.

IV. EXPERIMENTAL ANALYSIS AND RESULTS

The performance of the malicious URL detection system was evaluated using a confusion matrix, which provides a detailed breakdown of true and predicted classifications across the four URL categories: benign**,** defacement**,** malware**,** and phishing**.** The confusion matrix allows us to understand not only the overall accuracy but also how well each class is being correctly identified or misclassified.

A. Confusion Matrix for malicious URL detection

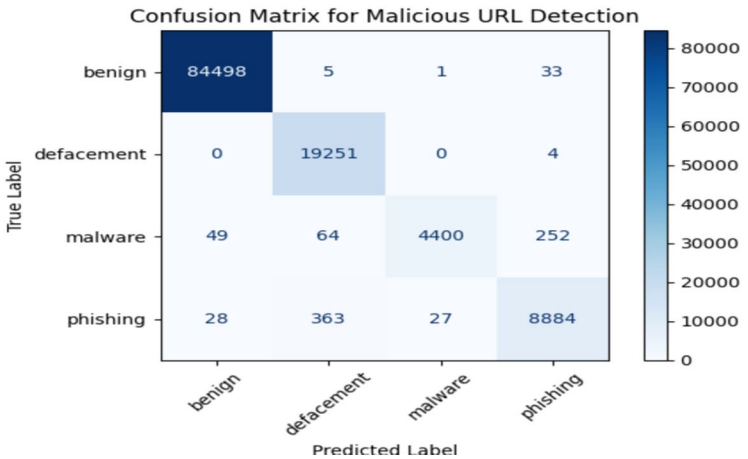


Figure-2:Confusion matrix for malicious URL Detection

1) Exceptional Performance on Benign URLs
- The model correctly classified 84,498 benign URLs with minimal errors.
- Only 39 misclassifications occurred (5 as defacement, 1 as malware, and 33 as phishing), demonstrating near-perfect precision (99.95%) for benign samples.

2) Strong Detection of Defacement URLs
- 19,251 defacement URLs were correctly identified.
- Only 4 instances were misclassified (likely as phishing), indicating high reliability in detecting defacement attacks.

3) Challenges in Malware Detection
- While **4,**400 malware URLs were correctly classified, 365 were misclassified (49 as benign, 64 as defacement, and 252 as phishing).
- The higher misclassification rate suggests malware URLs may share structural similarities with benign or phishing URLs, making them harder to distinguish.

4) Phishing Detection with Moderate Errors
- 8,884 phishing URLs were correctly identified.
- 418 misclassifications occurred (28 as benign, 363 as defacement, and 27 as malware).
- The confusion with defacement (363 errors) implies some phishing URLs exhibit features overlapping with defacement patterns.

B. Performance Metrics

Class	Precision	Recall	F1-Score	Support	Misclassified
Benign	99.90%	99,95%	99,93%	84,537	39
Defacement	99.80%	99.98%	99.89%	19,255	4
Malware	99.93%	92.34%	95.98%	4,765	365
Phishing	96.80%	95.52%	96.15%	9,302	418
Overall	99.31% Accuracy			117,859	826 Total Errors

Table 2:Performance Metrics

1) Precision (Positive Predictive Value):
- Highest for Malware (99.93%) and lowest for Phishing (96.80%), indicating rare false positives for malware but more for phishing.

2) Recall (Sensitivity):
- Near-perfect for Benign (99.95%) and Defacement (99.98%), but lower for Malware (92.34%), suggesting the model misses ~7.7% of malware URLs.

3) F1-Score:
- Balanced metric showing Benign and Defacement dominate (F1 > 99%), while Phishing and Malware trail slightly (96.15%, 95.98%).

4) Support:
- Reflects class imbalance (Benign dominates with 84,537 samples).

5) Misclassified:
- Malware and Phishing account for 94.8% of all errors (783/826), highlighting their comparative difficulty.
- Evaluation Criteria

C. Evaluation Criteria

The performance of the model is evaluated against parameters like Accuracy, Precision, Recall and F-Score. The results are as depicted.

Model	Accuracy	Precision	Recall	F1-Score	Comments
Random Forest	0.91	0.91	0.91	0.91	High accuracy and balanced performance; great for structured feature data.
XGBoost	0.908	0.91	0.91	0.91	Similar to Random Forest; efficient and good for handling class imbalance.
Decision Tree	0.9099	0.91	0.91	0.91	Simple, fast, and interpretable; slightly lower generalization ability.
Char-CNN	0.9931	0.985	0.969	0.976	Best overall; excels in learning from raw URLs but needs more compute power.

Table-3: Results

D. Output Screen

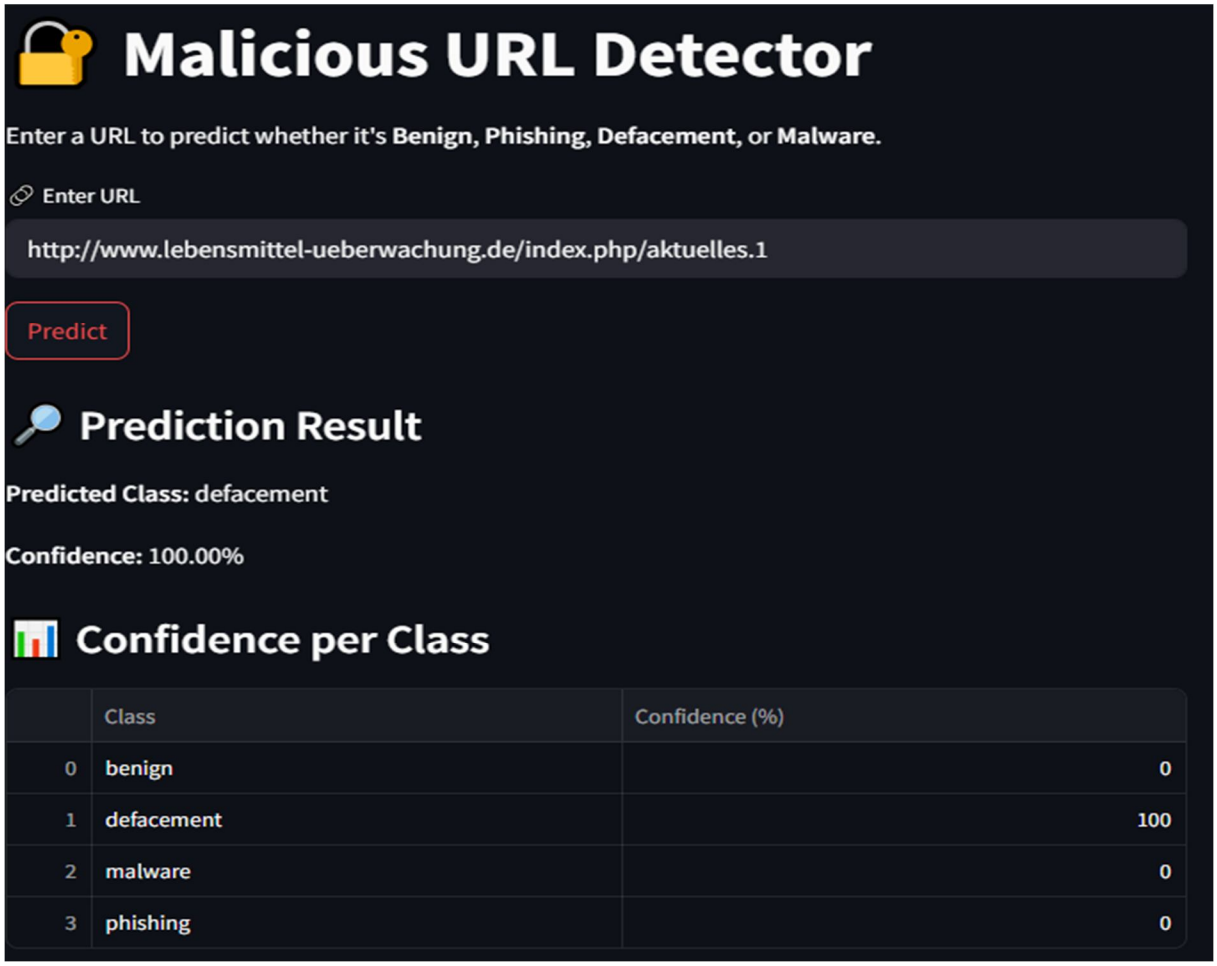


Figure 4:Output Screen

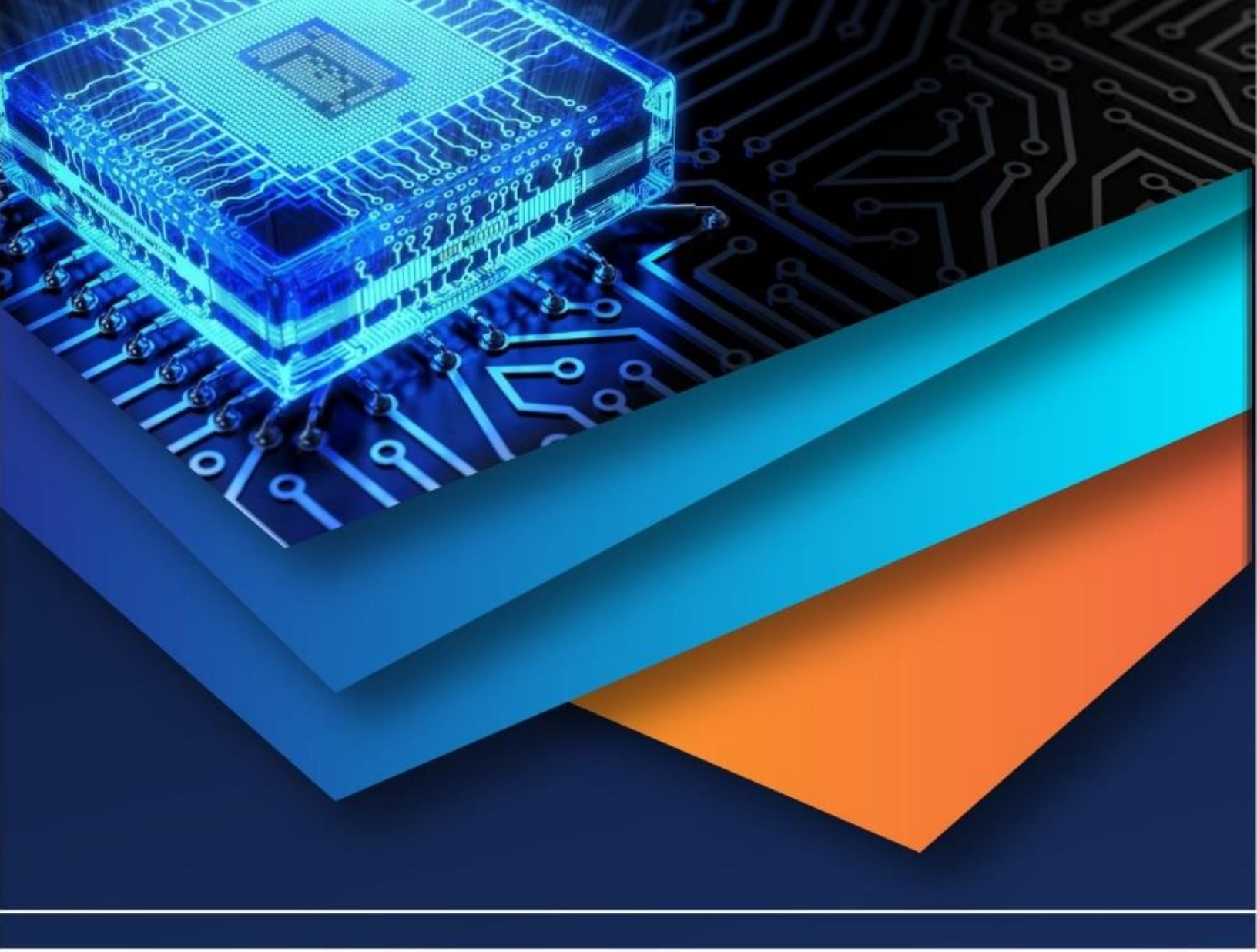
This output screen shows that the Char-CNN based Malicious URL Detector classified the input URL asbenign with 100%confidence. The confidence levels for other classes—defacement, malware, and phishing—are all **0%**, indicating high certainty in the benign classification

V. CONCLUSION

The rise in web-based services has also brought a significant increase in cyber threats, particularly through malicious URLs used for phishing, malware, and defacement. Traditional detection methods like blacklists and rule-based systems struggle to keep up with evolving threats. To overcome these limitations, this project developed a hybrid malicious URL detection system combining machine learning (ML) and deep learning (DL) techniques.The goal was to build a multi-class classifier to detect benign, phishing, malware, and defacement URLs with high accuracy. The system uses classical ML algorithms such as Decision Tree, Random Forest, and XGBoost, along with a Character-level Convolutional Neural Network (Char-CNN) that processes raw URLs directly.Lexical features were extracted from the URLs—such as length, special character counts, HTTPS usage, and presence of IP addresses. These features helped the ML models classify URLs effectively. Among the ML models, Random Forest and XGBoost performed best, with XGBoost offering the highest precision and scalability.The Char-CNN model eliminated the need for manual feature extraction and showed exceptional accuracy (over 99%) in detecting obfuscated and novel URLs. The combination of ML and DL enabled the system to balance speed, accuracy, and adaptability.Model performance was evaluated using accuracy, precision, recall, and F1-score. The ML models achieved over 91% accuracy, while the Char-CNN model surpassed 99%. The models were trained and tested on real-world data from Kaggle and exported using Joblib and TensorFlow, making them ready for real-time applications like browser plugins, email filters, or network firewall

REFERENCES

[1] Sahoo, Doyen, Chenghao Liu, and Steven CH Hoi. "Malicious URL detection using machine learning: a survey. CoRR abs/1701.07179 (2017)." (2017).
[2] O'Gorman, Brigid, et al. "Internet security threat report." A Report published by SYMANTEC 24 (2019): 32.
[3] Khonji, Mahmoud, Youssef Iraqi, and Andrew Jones. "Phishing detection: a literature survey." IEEE Communications Surveys & Tutorials 15.4 (2013): 2091-2121.
[4] Ma, Justin, et al. "Learning to detect malicious urls." ACM Transactions on Intelligent Systems and Technology (TIST) 2.3 (2011): 1-24.
[5] Choi, Hyunsang, Bin B. Zhu, and Heejo Lee. "Detecting malicious web links and identifying their attack types." 2nd USENIX Conference on Web Application Development (WebApps 11). 2011.
[6] Zhiwang, Cen, Xu Jungang, and Sun Jian. "A multi-layer bloom filter for duplicated URL detection." 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE). Vol. 1. IEEE, 2010.
[7] Sheng, Steve, Brad Wardman, Gary Warner, Lorrie Cranor, Jason Hong, and Chengshan Zhang. "An empirical analysis of phishing blacklists." (2009).
[8] Mohammad, Rami M., Fadi Thabtah, and Lee McCluskey. "An assessment of features related to phishing websites using an automated technique." 2012 international conference for internet technology and secured transactions. IEEE, 2012.
[9] Zhao, Peilin, and Steven CH Hoi. "Cost-sensitive online active learning with application to malicious URL detection." Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. 2013.
[10] Kinder, J., Katzenbeisser, S., Schallhart, C. and Veith, H., 2005, July. "Detecting malicious code by model checking." In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (pp. 174-187). Springer, Berlin, Heidelberg.
[11] Ma, Justin, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. "Identifying suspicious URLs: an application of large-scale online learning." In Proceedings of the 26th annual international conference on machine learning, pp. 681-688. 2009.
[12] Garera, Sujata, et al. "A framework for detection and measurement of phishing attacks." Proceedings of the 2007 ACM workshop on Recurring malcode. 2007.
[13] Patil, Dharmaraj R., and Jayantro B. Patil. "Malicious URLs detection using decision tree classifiers and majority voting technique." Cybernetics and Information Technologies 18, no. 1 (2018): 11-29.
[14] Vinayakumar, R., K. P. Soman, and Prabaharan Poornachandran. "Evaluating deep learning approaches to characterize and classify malicious URL's." Journal of Intelligent & Fuzzy Systems 34.3 (2018): 1333-1343.
[15] Darling, Michael, Greg Heileman, Gilad Gressel, Aravind Ashok, and Prabaharan Poornachandran. "A lexical approach for classifying malicious URLs." In 2015 international conference on high performance computing simulation (HPCS), pp. 195-202. IEEE, 2015.
[16] Menon, R. R., Kaartik, J., Nambiar, E. K., TK, A. K., Kumar, A. (2020, June). "Improving ranking in document based search systems." 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184) (pp. 914-921). IEEE.
[17] Menon, R.R.K., Akhil Dev, R., Bhattathiri, S.G., "An insight into the relevance of word ordering for text data analysis." 2020 fourth international conference on computing methodologies and communication (ICCMC). IEEE, 2020.
[18] Srinivasan, S., Vinayakumar, R., Arunachalam, A., Alazab, M., Soman, K. P. (2021). "DURLD: Malicious URL detection using deep learning-based character level representations." Malware analysis using artificial intelligence and deep learning, 535-554.
[19] Vazhayil, Anu, R. Vinayakumar, and K. P. Soman. "Comparative study of the detection of malicious URLs using shallow and deep networks." 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2018



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY