Islamic Azad University , Shiraz Branch
نشریه تحلیل مدارها، داده ها و سامانه ها
Journal of Circuits, Data and Systems Analysis
sanad.iau.ir/journal/jcdsa

Improving detection of phishing websites using machine learning and deep learning models

Hadi Rahim Beigi Chegini[1], Elham Parvinnia[2*]

[1] Department of Computer Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran
hadirahimbeigi@gmail.com
[2] Department of Computer Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran
eparvinnia@gmail.com

Abstract: Phishing attacks represent a significant cybersecurity threat, targeting internet users to steal confidential information. This research presents a hybrid deep learning model that employs data preprocessing, data balancing via SMOTE, and dimensionality reduction via PCA. The primary innovation of this model lies in its integrated approach, combining advanced techniques for data balancing, dimensionality reduction, and feature selection. This integration successfully addresses common challenges associated with imbalanced datasets and enhances overall model accuracy. he utilized dataset comprises key website features. Following data preprocessing, feature selection, and dimensionality reduction, several models -including Decision Tree, k-Nearest Neighbors, and Random Forest- were implemented. To mitigate class imbalance, techniques such as the Synthetic Minority Over-sampling Technique (SMOTE), Adaptive Synthetic Sampling (ADASYN), and Random Over-sampling were applied. Furthermore, feature selection methods based on Information Gain and dimensionality reduction were used to optimize model efficiency. Experimental results demonstrate that the proposed hybrid models achieve high accuracy in detecting phishing websites. Notably, the proposed Recurrent Neural Network (RNN) model attained 99% accuracy in identifying phishing websites using cross-validation, outperforming traditional methods.

Keywords: Phishing, machine learning, deep learning, feature selection, website classification.

JCDSA, Vol. 3, No. 3, Autumn 2025	Online ISSN: 2981-1295	Journal Homepage: https://sanad.iau.ir/en/Journal/jcdsa
Received: 2025-05-15	Accepted: 2025-11-24	Published: 2025-12-21

CITATION

Chegini, H., Parvinnia, E., *"Improving detection of phishing websites using machine learning and deep learning models"*, Journal of Circuits, Data and Systems Analysis (JCDSA), Vol. 3, No. 3, pp. 28-40, 2025.
DOI: 10.82526/jcdsa.2026.1206948

* Corresponding author

Extended Abstract

1- Introduction

Due to the rapid growth of the internet and online transactions, phishing attacks have become a significant cybersecurity threat. These attacks deceive users into revealing sensitive information by impersonating legitimate websites. Traditional methods like blacklists and manual rules have become ineffective against the increasing sophistication and volume of these attacks. Therefore, there is a pressing need for intelligent, automated models that can accurately identify phishing websites. This research addresses this challenge by proposing a hybrid framework that combines machine learning, deep learning, and ensemble learning to detect phishing websites more effectively. The framework tackles key challenges such as imbalanced datasets and the high dimensionality of website features, which are common in this field.

This study introduces several innovations to enhance phishing detection. First, it addresses the data imbalance problem by employing oversampling techniques like SMOTE and ADASYN, combined with Principal Component Analysis (PCA) for dimensionality reduction. This approach improves classification model performance while reducing computational complexity. Second, the research optimizes a Recurrent Neural Network (RNN) model, which achieves a remarkable accuracy of 99%—outperforming other models such as Decision Trees, SVM, and Random Forests. The combination of these advanced data preprocessing and modeling techniques provides a robust and efficient solution for accurately identifying and mitigating phishing threats in an evolving online landscape.

2- Methodology

The proposed phishing detection method employs a nine-step process to enhance model accuracy and interpretability. Initially, the system preprocesses the Phishing Website Dataset, which includes numerical and binary features, by cleaning and scaling the data and removing irrelevant features through correlation analysis. To address the class imbalance—where legitimate websites far outnumber phishing ones—the dataset is balanced using three techniques: SMOTE, ADASYN, and RandomOverSampler. The paper reports a significant improvement in class distribution, with RandomOverSampler achieving a perfect balance. Feature selection is then performed using Information Gain, Gain Ratio, and Principal Component Analysis (PCA) to identify the most critical features. The data is then split into training and test sets, with an 80/20 ratio, using stratified sampling to maintain class proportions. This comprehensive data preparation ensures a robust foundation for the subsequent modeling phases.

The core of the research involves implementing and comparing various machine learning and deep learning models. Specifically, the study uses Ensemble Learning and Machine Learning models like Support Vector Machine, Decision Tree, Random Forest, and XGBoost, alongside Deep Learning models such as LSTM, GRU, and RNN. These models are meticulously optimized, with the RNN model specifically fine-tuned by increasing neurons and reducing the learning rate to achieve a high accuracy of 99%. The performance of all models is evaluated using 10-Fold Cross-Validation to prevent overfitting and ensure reliable, generalizable results. Finally, the SHAP method is applied to interpret the deep learning models, moving beyond the "black box" nature of these complex algorithms. By calculating the contribution of each feature to the model's predictions, SHAP helps identify which attributes, such as *SSLfinal_State* or *URL_of_Anchor*, are most influential in classifying a website as phishing. This interpretability adds transparency and trust, which is crucial in cybersecurity.

3- Results and discussion

The analysis compares the performance of several machine learning, ensemble learning, and deep learning models for phishing detection. The “Recurrent Neural Network (RNN)” emerged as the top performer, achieving an impressive accuracy of 99.01%, with XGBoost and Random Forest also showing strong results. In contrast, models like Decision Tree and AdaBoost had weaker performance. To ensure the reliability of these findings, a “10-fold cross-validation” was conducted, which confirmed the RNN's stability with an average accuracy of “98.74% ± 0.32%”, and its superior performance over LSTM and GRU models. The results highlight that the deep learning approach, particularly the RNN, is highly effective for this task, outperforming traditional machine learning methods in both accuracy and stability.

The “SHAP” analysis revealed that features such as *having_IP_Address*, *Shortining_Service*, and *URL_Length* were the strongest indicators for classifying a website as phishing. Conversely, features like a valid *SSLfinal_State* and *Domain_registeration_length* were the most influential in classifying a website as legitimate. This interpretability provides valuable insights into how the model works, thereby increasing trust in its predictions.

4- Conclusion

This study proposes a hybrid approach to detect phishing attacks.The methodology included crucial preprocessing steps such as SMOTE and ADASYN for data balancing, and Information Gain and PCA for feature selection. The analysis of various models, including SVM, Random Forest, LSTM, and RNN, revealed that the RNN model achieved the highest accuracy at 99%, outperforming all other models. The research confirmed that deep learning models are more effective for this task, though they face challenges like high computational costs and reduced interpretability compared to traditional machine learning methods.

Islamic Azad University , Shiraz Branch
نشریه تحلیل مدارها، داده ها و سامانه ها
Journal of Circuits, Data and Systems Analysis
sanad.iau.ir/journal/jcdsa

بهبود تشخیص وب‌سایت‌های فیشینگ با استفاده از مدل‌های یادگیری ماشین و یادگیری عمیق

هادی رحیم بیگی چگینی [۱]، الهام پروین‌نیا [۲*]

۱- گروه مهندسی کامپیوتر، واحد شیراز، دانشگاه آزاد اسلامی، شیراز، ایران (hadirahimbeigi@gmail.com)
۲- گروه مهندسی کامپیوتر، واحد شیراز، دانشگاه آزاد اسلامی، شیراز، ایران (eparvinnia@gmail.com)

چکیده: حملات فیشینگ از تهدیدات مهم امنیت سایبری محسوب می‌شوند که کاربران اینترنت را هدف قرار داده و اطلاعات محرمانه آن‌ها را سرقت می‌کنند. در این پژوهش، یک مدل ترکیبی مبتنی بر یادگیری عمیق با به‌کارگیری پیش‌پردازش داده‌ها، متعادل‌سازی داده‌ها و کاهش ابعاد ارائه شده‌است. نوآوری اصلی این مدل، در ترکیب همزمان چندین تکنیک پیشرفته برای متعادل‌سازی داده‌ها، کاهش ابعاد و انتخاب ویژگی‌ها نهفته است که توانسته مشکلات رایج در داده‌های نامتوازن را برطرف کند و دقت مدل را به‌طور چشمگیری افزایش دهد. مجموعه‌داده مورد استفاده، شامل ویژگی‌های کلیدی وب‌سایت‌ها است. پس از اعمال مراحل پیش‌پردازش داده‌ها، انتخاب ویژگی و کاهش ابعاد، مدل‌های مختلفی از جمله درخت تصمیم، k- نزدیک‌ترین همسایه و جنگل تصادفی پیاده‌سازی شدند. به‌منظور متعادل‌سازی توزیع کلاس‌ها، از روش‌هایی همچون افزایش نمونه‌سازی مصنوعی اقلیت، نمونه‌سازی مصنوعی تطبیقی و افزایش نمونه‌برداری تصادفی استفاده شده‌است. همچنین، روش‌های انتخاب ویژگی مبتنی بر کسب اطلاعات و کاهش ابعاد به‌کار گرفته شدند تا کارایی مدل‌ها بهبود یابد. نتایج آزمایش‌ها نشان می‌دهد که مدل‌های ترکیبی ارائه‌شده در این پژوهش، از دقت بالایی در تشخیص وب‌سایت‌های فیشینگ برخوردار هستند. مدل شبکه عصبی بازگشتی پیشنهادی با استفاده از روش اعتبارسنجی متقابل به دقت ۹۹٪ در شناسایی این وب‌سایت‌ها دست یافته‌است که در مقایسه با روش‌های سنتی، عملکرد برتر و کارآمدتری را نشان می‌دهد.

واژه‌های کلیدی: فیشینگ، یادگیری ماشین، یادگیری عمیق، انتخاب ویژگی، طبقه‌بندی وب‌سایت‌ها.

DOI: 10.82526/jcdsa.2026.1206948

نوع مقاله: پژوهشی
تاریخ ارسال مقاله: ۱۴۰۴/۰۲/۲۵ تاریخ پذیرش مقاله: ۱۴۰۴/۰۹/۰۳ تاریخ چاپ مقاله: ۱۴۰۴/۰۹/۳۰

۱- مقدمه

با رشد سریع فناوری‌های اینترنتی و افزایش تراکنش‌های آنلاین، حملات فیشینگ به یکی از تهدیدات جدی در حوزه امنیت سایبری تبدیل شده‌اند. این حملات با تقلید از وب‌سایت‌های معتبر، کاربران را فریب داده و اطلاعات حساس مانند نام کاربری، رمز عبور و اطلاعات بانکی آن‌ها را سرقت می‌کنند. براساس گزارش‌های امنیتی، میزان حملات فیشینگ در سال‌های اخیر به‌طور چشمگیری افزایش یافته و روش‌های سنتی مبتنی بر فهرست سیاه [۲] و قوانین دستی [۳] کارایی خود را در مقابله با این حملات از دست داده‌اند. بنابراین، توسعه مدل‌های هوشمند و خودکار که بتوانند وب‌سایت‌های فیشینگ را با دقت بالا شناسایی کنند، به یک نیاز اساسی تبدیل شده است [۱]. یکی از روش‌های موثر در تشخیص حملات فیشینگ، استفاده از الگوریتم‌های یادگیری ماشین [۴] و یادگیری عمیق [۵] و یادگیری گروهی [۶] است. این الگوریتم‌ها با تحلیل ویژگی‌های مختلف یک وب‌سایت، می‌توانند به‌طور خودکار تفاوت بین وب‌سایت‌های قانونی و فیشینگ را شناسایی کنند. در این پژوهش، یک چارچوب ترکیبی مبتنی بر روش‌های یادگیری ماشین و یادگیری عمیق و یادگیری گروهی ارائه شده است که با استفاده از تکنیک‌های انتخاب ویژگی، کاهش ابعاد و متعادل‌سازی کلاس‌ها، عملکرد مدل‌های طبقه‌بندی را بهبود می‌بخشد. چالش‌های اصلی در تشخیص فیشینگ به صورت زیر است:

عدم تعادل در کلاس‌های داده: معمولاً تعداد وب‌سایت‌های قانونی بسیار بیشتر از وب‌سایت‌های فیشینگ است، که باعث می‌شود مدل‌های یادگیری به سمت طبقه اکثریت (وب‌سایت‌های قانونی) تمایل پیدا کنند. برای حل این مشکل، از روش‌های افزایش نمونه‌برداری

* نویسنده مسئول

[2] Blacklist
[3] Rule-Based
[4] Machine Learning
[5] Deep Learning
[6] Ensemble Learning

مصـــنوعی[1]، روش تطبیقی افزایش نمونـه‌برداری[2] و نمونـه‌برداری تصادفی[3] جهت متعادل‌سازی داده‌ها استفاده شده است [۲].

ویژگی‌های متنوع وب‌ســـایت‌ها: ویژگی‌های مبتنی بر آدرس[4]، ویژگی‌های HTML و جاوا اســـکریپت، و ویژگی‌های مرتبط با گواهی امنیتی[5] در تشخیص فیشینگ تأثیرگذار هستند. استفاده از روش‌های انتخاب ویژگی مبتنی بر میزان اطلاعات[6] و کاهش ابعاد با روش تحلیل مؤلفه‌های اصلی[7] می‌تواند به بهبود عملکرد مدل کمک کند.

دقت و کارایی مدل‌ها: برخی از مدل‌های یادگیری ماشـــین مانند درخت تصـــمیم[8] عملکرد خوبی دارند، اما ممکن اســـت در مواجهه با مجموعه داده‌های بزرگ و متنوع کارایی مناسـبی نداشـته باشـند. از این‌رو، در این پژوهش علاوه بر مدل‌های یادگیری ماشـین، از مدل‌های یادگیری عمیق نیز استفاده شده است تا ساختارهای پیچیده‌تر را یاد بگیرند و دقت پیش‌بینی را افزایش دهند.

در این پژوهش، یک سـیسـتم ترکیبی مبتنی بر یادگیری گروهی، یادگیری ماشین و یادگیری عمیق برای تشخیص خودکار وب‌سایت‌های فیشـینگ ارائه شـده است. نوآوری‌های این پژوهش شـامل اسـتفاده از روش‌های مختلف متعادل‌ســـازی داده‌ها برای بهبود عملکرد مدل‌های طبقه‌بندی، بهره‌گیری از انتخاب ویژگی و کاهش ابعاد برای اســـتخراج مهم‌ترین ویژگی‌های مؤثر در تشـخیص فیشـینگ و مقایسـه جامع بین مدل‌های یادگیری گروهی، یادگیری ماشـــین و یادگیری عمیق برای یافتن بهترین مدل می باشـد. در این پژوهش، نوآوری مدل پیشـنهادی به دو بخش اصلی تقسیم می‌شود:

ترکیب SMOTE و PCA: اســـتفاده از روش‌های SMOTE و ADASYN برای متعادل‌سازی داده‌ها در کنار استفاده از PCA برای کاهش ابعاد داده‌ها، در مدل‌های یادگیری ماشـــین و یادگیری عمیق، باعث بهبود عملکرد مدل‌ها شـده اسـت. این رویکرد نه تنها پیچیدگی محاسباتی را کاهش داده بلکه کارایی مدل‌ها را افزایش داده است.

بهینه‌سـازی مدل شـبکه عصـبی بازگشـتی (RNN): در این پژوهش، مدل RNN بهینه‌ســـازی شـــده اســـت تا دقت بیشـــتری در شناسایی الگوهای فیشینگ در داده‌های پیچیده بدست آورد. این مدل توانسـته اسـت به دقت ۹۹٪ برسـد که به‌طور قابل‌ملاحظه‌ای عملکرد بالاتری نسبت به سایر مدل‌ها از جمله درخت تصمیم، SVM و جنگل تصادفی نشان داده است.

این مقاله به‌صـــورت زیر ســـازمان‌دهی شـــده اســـت: بخش دوم، مطالعات پیشـین و روش‌های موجود برای تشـخیص فیشـینگ بررسـی می‌شـوند. در بخش سـوم، روش‌شـناسـی پژوهش شـامل مجموعه داده، پیش‌پردازش، متعادل‌سـازی داده‌ها و روش‌های انتخاب ویژگی تشـریح می‌شـود. در بخش چهارم، نتایج آزمایش‌ها و مقایسـه عملکرد مدل‌های مختلف ارائـه می‌شـــود. در بخش پنجم، جمع‌بنـدی، نتیجـه‌گیری و پیشنهادهایی برای کارهای آینده بیان می‌شود..

۲- مروری بر کارهای مرتبط

با افزایش حملات فیشـــینگ و پیچیدگی تکنیک‌های فریب کاربران، روش‌های مختلفی برای تشـــخیص و مقابله با این تهدیدات پیشـــنهاد شـده‌اند. در این بخش، به بررسـی مطالعات مرتبط پرداخته و نقاط قوت و ضعف روش‌های موجود را تحلیل می‌کنیم.

۲-۱- روش‌های سنتی تشخیص فیشینگ

در مراحل اولیه، روش‌های مبتنی بر فهرست سـیاه و فهرسـت سـفید[9] برای شـناسـایی وب‌سـایت‌های فیشـینگ توسـعه یافتند. در این روش‌ها، فهرســـتی از آدرس های اینترنتی مخرب یا آدرس‌های امن نگهداری شده و هر URL جدید با این فهرست مقایسه می‌شود.

نقاط قوت: سرعت بالا در شناسایی وب‌سایت‌های شناخته‌شده، سادگی پیاده‌سازی و کارایی در مقیاس کوچک

نقاط ضعف: عدم توانایی در شـناسـایی حملات فیشـینگ جدید، نیاز به بروزرســـانی مداوم پایگاه داده، آســـیب‌پذیری در برابر حملات تغییر سریع URL

مطالعاتی مانند [۳] نشان داده‌اند که روش‌های مبتنی بر فهرست سیاه تنها ۵۰ تا ۷۰ درصـد از وب‌سـایت‌های فیشـینگ جدید را شـنـاسـایی می‌کنند، که این میزان برای مقابله با تهدیدات مدرن کافی نیست.

۲-۲- روش‌های مبتنی بر یادگیری ماشین

با پیشـرفت یادگیری ماشـین، مدل‌هایی مانند ماشـین بردار پشـتیبان[10]، درخت تصـمیم، جنگل تصـادفی[11]، نزدیکترین همسـایه[12] و روشـهای تقویت مدل[13] برای تشخیص فیشینگ مورد استفاده قرار گرفته‌اند. این مدل‌ها با تحلیل ویژگی‌های مختلف وب‌ســـایت‌ها، الگوهای پنهان را شـناسـایی کرده و تصـمیم‌گیری انجام می‌دهند. مقاله‌ی [۴] نشـان داد که مدل XGBoost با دقت ۹۵٪ در تشـــخیص فیشـــینگ عملکرد بسـیار بهتری نسـبت به مدل‌های سـنتی دارد. مطالعه دیگری نشـان داد که مدل Random Forest با اسـتفاده از ۵۰ ویژگی منتخب، توانسـت به دقت ۹۳٪ دست یابد [۵].

محـدودیـت‌هـای روش‌هـای یـادگیری مـاشـــین: دقـت پـایین در ویژگی‌های غیر عددی و متنی حســـاســـیت به عدم تعادل داده‌ها (داده‌های فیشـینگ معمولاً کمتر از داده‌های غیر فیشـینگ هسـتند)، عدم توانایی در مدل‌سازی توالی‌های پیچیده و وابستگی زمانی مطالعات

[1] SMOTE
[2] ADASYN
[3] RandomOverSampler
[4] URL
[5] SSL/TLS
[6] Information Gain
[7] PCA
[8] Decision Tree
[9] Whitelist
[10] SVM
[11] Random Forest
[12] KNN
[13] XGBoost

نشـــان داده‌اند که ترکیب روش‌های ML با کاهش ابعاد داده‌ها مانند PCA و Information Gain می‌تواند منجر به بهبود دقت شـــود، اما همچنان ضعف‌هایی در تحلیل الگوهای پیچیده باقی می‌ماند.

۲-۳- روش‌های مبتنی بر یادگیری عمیق

با ظهور یادگیری عمیق، مدل‌هایی مانند شبکه‌های عصبی بازگشتی [1]، حافظه بلندمدت [2] و واحدهای دروازه‌ای بازگشـــتی [3] برای شـــناســـایی الگوهای پیچیده در داده‌های فیشینگ مورد استفاده قرار گرفته‌اند. این روش‌ها قابلیت یادگیری وابســـتگی‌های طولانی‌مدت و کشـــف الگوهای پنهان در داده‌ها را دارند. در [۷]، مدل LSTM توانســـت با دقت ۹۷٪ عملکرد بسیار بهتری نسبت به مدل‌های یادگیری ماشین نشان دهد. تحقیقی دیگر نشـــان داد که ترکیب CNN و LSTM باعث بهبود ویژگی‌برداری و افزایش دقت مدل تا ۹۸٪ شد [۸].

محدودیت‌های روش‌های یادگیری عمیق: زمان آموزش طولانی و نیاز به قدرت پردازشـــی بالا، حســـاســـیت به عدم تعادل داده‌ها، نیاز به مقدار زیادی داده آموزشـــی برای رســـیدن به دقت بالا. با این حال، مطالعات اخیر نشـــان داده‌اند که اســـتفاده از تکنیک‌های بهینه‌ســـازی، تنظیم پارامترهای شـــبکه و ترکیب روش‌های مختلف می‌تواند عملکرد مدل‌های یادگیری عمیق را بهبود بخشد.

۲-۴- ترکیب یادگیری ماشین و یادگیری عمیق [4]

برخی از مطالعات اخیر ترکیبی از روش‌های ML و DL را برای بهبود عملکرد تشـــخیص فیشـــینگ پیشـــنهاد داده‌اند. در این روش‌ها ابتدا ویژگی‌های کلیدی با اســـتفاده از مدل‌های یادگیری ماشـــین اســـتخراج می‌شـــوند. ســـپس این ویژگی‌ها به مدل‌های یادگیری عمیق مانند LSTMو GRU وارد می‌شـــوند تا وابســـتگی‌های زمانی و الگوهای پیچیده شـــناســـایی شـــوند. در [۹]، ترکیب جنگل تصـــادفی و LSTM باعث شد که دقت مدل به ۹۹٪ برسد.

مزایای روش‌های ترکیبی: بهبود دقت مدل با ترکیب قدرت تحلیل ویژگی‌های یادگیری ماشـــین و یادگیری توالی‌های پیچیده توســـط یادگیری عمیق، کاهش زمان پردازش در مقایســـه با اســـتفاده صـــرف از یادگیری عمیق، قابلیت شناسایی الگوهای پیچیده و جدید

۲-۵- مقایسه مطالعات پیشین با پژوهش حاضر

با توجه به مرور مطالعات گذشـــته، محدودیت‌هایی مانند عدم تعادل داده‌ها، وابســـتگی به ویژگی‌های خاص، و نیاز به قدرت پردازشـــی بالا همچنان در مدل‌های موجود مشـــاهده می‌شـــود. در ســـال‌های اخیر (از ۲۰۲۰ به بعد)، تمرکز پژوهش‌ها در تشخیص فیشـــینگ وب‌سایت‌ها از روش‌های مبتنی بر ویژگی‌های دســـتی و مدل‌های کلاســـیک به ســـمت مـعـمـاری‌هـای عـمـیـق، مـدل‌هـای تـرکـیـبـی و چـارچـوب‌هـای تفســـیرپذیر (XAI) تغییر کرده اســـت. بخشـــی از تحقیقات جدید بر اســـتفاده از شـــبکه‌های کانولوشـــنی و بازگشـــتی برای اســـتخراج همزمان الگوهای ســـاختاری و ترتیبی در URL یا ویژگی‌های وب تمرکز دارند. برای مثال، برخی مطالعات با به‌کارگیری معماری‌های CNN، LSTM یا ترکیب CNN–LSTM توانســـته‌اند عملکرد بالایی در تشـــخیص فیشـــینگ ارائه دهند و نشـــان دهند که شـــبکه‌های عمیق در اســـتخراج الگوهای پیچیده از داده‌ها مزیت دارند [۱۶].

در رویکردهای ترکیبی جدید، تلاش می‌شـــود با ادغام شـــبکه‌های عمیق (برای یـادگیری الگوهـای غیرخطی) بـا اجزای مکمـل (مـانـنـد انتخاب ویژگی، بهینه‌سازی فراپارامترها یا روش‌های بالانس‌سازی داده) دقت افزایش و خطاهای امنیتی) به‌ویژه False Negative کاهش یابد. نمونه‌ای از این روند، مدل‌های CNN–RNN اســـت که با هدف استفاده از نقاط قوت هر دو خانواده شبکه (ویژگی‌برداری کانولوشنی و یادگیری وابستگی‌های ترتیبی) پیشـنهاد شده‌اند [۱۷]. از سوی دیگر، مسیر مهم دیگر در پژوهش‌های جدید، حرکت به سمت «تفسیرپذیری» است؛ زیرا در امنیت سایبری، صرف دقت بالا کافی نیست و باید روشن باشد مدل بر اساس چه ویژگی‌هایی تصمیم گرفته است. در این راستا، چارچوب‌های مبتنی بر SHAP/LIME و انتخاب ویژگی تفســـیرپذیر رشـــد چشـــمگیری داشـــته‌اند و برخی کارها به‌صـــورت مســـتقیم نشـــان داده‌اند که ترکیب روش‌های XAI با مدل‌های یادگیری می‌تولند هم اعتمادپذیری و هم کارایی را ارتقا دهد[۱۸].

جایگاه و نوآوری پژوهش حاضـر نسـبت به مسـیرهای فوق در این است که به‌جای اتکا به یک معماری صرفاً عمیق یا صرفاً کلاسیک، یک زنجیره کامل و منســـجم از «پیش‌پردازش هدفمند» را به‌کار می‌گیرد: (۱) اســـتفـاده از تکنیـک‌هـای متعـادل‌ســـازی داده‌هـا (SMOTE، RandomOverSampler، ADASYN) برای بهبود عملکرد مدل‌های یـادگیری و گزارش توزیع کلاس قبـل/بعـد، (۲) انتخـاب ویژگی بـا معیارهای اطلاعاتی، (۳) کاهش بُعد با PCA برای حفظ بیشـــترین واریانس و کاهش هم‌بســـتگی، و (۴) مقایســـه نظام‌مند مدل‌هایML ، یادگیری گروهی و DL در یک چارچوب واحد. این ترکیب باعث کاهش پیچیدگی محاســـباتی و افزایش پایداری مدل شـــده و در نهایت، مدل RNNبهینه‌شده توانسته است دقت بالاتری نسبت به سایر مدل‌ها با کاهش نرخ خطای False Positive و False Negative توسط تنظیم بهینه پارامترهای مدل‌ها به‌دست آورد. در ادامه، روش‌شناسی پژوهش ارائه می‌شود که نحوه پیاده‌سازی مدل‌ها و تنظیمات را تشریح می‌کند.

۲-۶- جمع‌بندی این بخش

روش‌های سنتی فهرست سیاه و سفید ناکارآمد بوده و نیاز به روش‌های پیشـرفته‌تر وجود دارد. اگرچه یادگیری ماشـین دقت بالایی دارد اما در تشـــخیص الگوهای پیچیده محدود اســـت. همچنین یادگیری عمیق توانایی شـناسـایی وابستگی‌های پیچیده را دارد اما چالش‌هایی از جمله زمان پردازش بالا دارد. بنابراین ترکیب روش‌های یادگیری ماشـــین و

[1] RNN
[2] LSTM
[3] GRU
[4] Hybrid Models

یادگیری عمیق باعث افزایش دقت شده و پژوهش حاضر این رویکرد را بهینه‌سازی کرده است. در مرحله بعدی، بخش روش‌شناسی[1] ارائه خواهد شد.

۳- ارائه روش پیشنهادی

در این بخش، روش پیشنهادی برای تشخیص وب‌سایت‌های فیشینگ ارائه می‌شود. مدل پیشنهادی ترکیبی از یادگیری ماشین و یادگیری عمیق است که با استفاده از روش‌های متعادل‌سازی داده، کاهش ابعاد و بهینه‌سازی مدل‌ها دقت تشخیص را افزایش می‌دهد. در ادامه، مراحل اصلی پیاده‌سازی این روش توضیح داده می‌شود.

۳-۱- چارچوب کلی روش پیشنهادی

روش پیشنهادی از نه گام اصلی تشکیل شده است:

- گام ۱: دریافت و پیش‌پردازش داده‌ها.
- گام ۲: متعادل‌سازی مجموعه داده با روش‌های (SMOTE، ADASYN، RandomOverSampler).
- گام ۳: انتخاب ویژگی‌های مهم با استفاده از روش‌های Information Gain و PCA .
- گام ۴: تقسیم داده‌ها به مجموعه‌های آموزشی و آزمایشی.
- گام ۵: پیاده‌سازی مدل‌های یادگیری گروهی و یادگیری ماشین ازجمله ماشین بردار پشتیبان، درخت تصمیم، جنگل تصادفی.
- گام ۶: پیاده‌سازی مدل‌های یادگیری عمیق شامل LSTM، GRU و RNN.
- گام ۷: تنظیم بهینه پارامترها برای افزایش دقت مدل‌ها و بهینه‌سازی RNN با افزایش نورون‌ها و کاهش نرخ یادگیری برای دستیابی به دقت بالا.
- گام ۸: ارزیابی عملکرد مدل‌ها، استفاده از روش اعتبارسنجی متقابل و مقایسه نتایج.
- گام ۹: استفاده از روش SHAP برای تفسیر مدل‌های یادگیری عمیق و مشخص نمودن تاثیر هر ویژگی‌ها در عملکرد مدل.

در ادامه، هر یک از این مراحل به تفصیل توضیح داده می‌شود.

۳-۲- دریافت و پیش‌پردازش داده‌ها

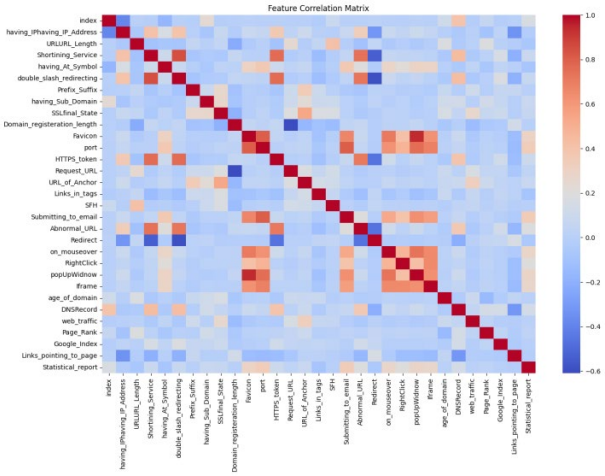
مجموعه داده: این پژوهش از مجموعه داده Phishing Website Dataset که در تحقیقات پیشین برای تشخیص حملات فیشینگ استفاده شده است، بهره می‌برد. این مجموعه شامل ویژگی‌های عددی و باینری است که مشخصه‌های مهم وب‌سایت‌ها را نشان می‌دهند. این مجموعه داده دارای ۱۱,۰۵۵ نمونه بوده و برچسب خروجی آن دودویی (فیشینگ/قانونی) است. تعداد ویژگی‌های ورودی (پس از حذف ستون‌های شاخص/شناسه) ۳۰ ویژگی است. ویژگی‌ها عمدتاً به‌صورت عددی کدگذاری شده‌اند و بخش از آن‌ها ماهیت دودویی/سه‌حالته دارند (برای نمونه مقادیر ۱-، ۰ و ۱ برای نمایش وضعیت‌های متفاوت یک ویژگی). در این مجموعه‌داده مقدار گمشده گزارش نشده و داده ها برای تحلیل‌های یادگیری ماشین مناسب هستند [۱۰].

محدودیت‌های مجموعه‌داده: با وجود کاربرد گسترده این مجموعه‌داده در پژوهش‌ها، چند محدودیت مهم وجود دارد: ویژگی‌ها عمدتاً «ایستا» هستند و برخی ویژگی‌های رفتاری یا مبتنی بر زمان (مانند تغییرات لحظه‌ای دامنه یا کمپین‌های زودگذر فیشینگ) را پوشش نمی‌دهند. داده از محیط واقعیِ آنلاین به‌صورت جریان[2] استخراج نشده است؛ بنابراین برای استقرار عملی، لازم است روش پیشنهادی روی داده‌های به‌روز و عملیاتی نیز آزمون شود. کدگذاری ۱-/۰/۱ باعث می‌شود برخی الگوریتم‌ها نیازمند نرمال‌سازی دقیق باشند؛ به همین دلیل از مقیاس‌بندی استاندارد استفاده شد.

مراحل پیش‌پردازش داده: حذف مقادیر نامعتبر و داده‌های پرت؛ تبدیل مقادیر غیرعددی به مقادیر عددی؛ مقیاس‌بندی داده‌ها با استفاده از StandardScaler برای بهبود کارایی مدل‌ها؛ بررسی همبستگی ویژگی‌ها با استفاده از ماتریس همبستگی و حذف ویژگی‌های غیرمؤثر (شکل ۱).

جدول (۱): مشخصات مجموعه داده

مولفه	مقدار/توضیح
نام مجموعه‌داده	Phishing Websites Dataset
تعداد کل نمونه‌ها	11,055
تعداد وب‌سایت قانونی	6,157
تعداد وب‌سایت فیشینگ	4,898
تعداد ویژگی‌های ورودی	30
نوع ویژگی‌ها	عمدتاً عددیِ کدگذاری‌شده (باینری/سه‌حالته)
وجود مقدار گمشده	خیر
نوع برچسب خروجی	دودویی (فیشینگ/قانونی)
نمونه ویژگی‌ها	SSLfinal_State, URL_Length, Shortining_Service, having_IP_Address



شکل (۱): همبستگی ویژگی‌ها با ماتریس همبستگی

[1] Methodology

[2] real-time

۳-۳- متعادل‌سازی مجموعه داده

مشکل مجموعه داده: در داده‌های اولیه، نمونه‌های وب‌سایت‌های فیشینگ کمتر از وب‌سایت‌های قانونی هستند که باعث عدم تعادل کلاس‌ها شده و کارایی مدل را کاهش می‌دهد.

راهکار: به‌منظور بهبود عملکرد مدل‌ها، از سه تکنیک SMOTE، ADASYN و RandomOverSampler استفاده می‌شود [۱۱].

- SMOTE[1]: تولید نمونه‌های مصنوعی از داده‌های اقلیت
- ADASYN[2]: تمرکز بیشتر بر نمونه‌های سخت‌تر و ایجاد داده‌های جدید بر اساس آن‌ها
- RandomOverSampler: افزایش تعداد داده‌های کلاس اقلیت برای رسیدن به تعادل کامل

با استفاده از این روش‌ها، تعادل داده‌ها بهبود یافته و مدل‌ها دقت بیشتری در شناسایی حملات فیشینگ خواهند داشت. در جدول (۲)، مشخصات مجموعه داده قبل و بعد از متعادل‌سازی آورده شده است.

۳-۴- انتخاب ویژگی‌های مهم

در این پژوهش از سه روش انتخاب ویژگی شامل IG[3]، GR[2] و PCA[4] برای استخراج ویژگی‌های مهم استفاده شده است [۱۲]. هدف از این ترکیب، افزایش دقت مدل و کاهش پیچیدگی محاسباتی است. در این بخش، به‌طور دقیق‌تری بیان می‌شود که چرا این ترکیب از روش‌ها به‌طور همزمان استفاده شده است و چه تأثیری در عملکرد مدل‌ها داشته است:

جدول (۲): مشخصات مجموعه داده قبل و بعد از متعادل‌سازی

```
Class distribution before balancing:
Counter({1: 6157, 0: 4898})
Class distribution after SMOTE: Counter({1:
6157, 0: 5849})
ADASYN skipped as SMOTE already balanced the
data.
Class distribution after RandomOverSampler:
Counter({0: 6157, 1: 6157})
```

جدول (۳): محاسبه ویژگی های منتخب با IG,GR,PCA

ردیف	ویژگی	مقدار اطلاعات (IG)	نسبت بهره (GR)	اهمیت PCA (PCA Variance Ratio)
1	SSLfinal_State	0.373936	1.000000	1.000388
2	URL_of_Anchor	0.314312	0.840551	2.092429
3	having_Sub_Domain	0.099388	0.265790	0.855099
4	Prefix_Suffix	0.092233	0.246654	0.855187
5	web_traffic	0.091352	0.244298	0.647140
6	Request_URL	0.060253	0.161132	0.642662
7	Domain_registeration_length	0.055379	0.148097	1.522719
8	Links_in_tags	0.028632	0.076570	1.664032

- IG و GR: این دو روش بر اساس تحلیل اطلاعات متقابل و نسبت‌های اطلاعاتی ویژگی‌ها با متغیر هدف هستند که به شناسایی ویژگی‌های مؤثر و حذف ویژگی‌های غیرضروری کمک می‌کند.
- PCA: این روش به‌منظور کاهش ابعاد داده‌ها و حفظ بیشترین واریانس در داده‌ها مورد استفاده قرار گرفته است.

برای تحلیل بهتر، در این مقاله نتایج آزمایشات آورده شده است که نشان می‌دهد استفاده همزمان از این سه روش، به‌طور معناداری عملکرد مدل را بهبود بخشیده است. در جدول (۲)، محاسبه ویژگی‌های منتخب با IG، GR و PCA آورده شده است. در این جدول مشاهده می‌شود که ویژگی‌هایی مانند *SSLfinal_State* و *URL_of_Anchor* از بالاترین اهمیت برخوردار هستند و نقش تعیین‌کننده‌ای در تشخیص وب‌سایت‌های فیشینگ دارند.

۳-۵- تقسیم‌بندی داده‌ها

تقسیم‌بندی داده‌ها به دو مجموعه: از ۸۰٪ داده‌ها برای آموزش مدل‌ها و ۲۰٪ داده‌ها برای ارزیابی عملکرد مدل‌ها استفاده می‌شود.

نمونه‌گیری طبقه‌بندی‌شده[5]: برای حفظ نسبت کلاس‌های فیشینگ و غیر فیشینگ در مجموعه‌های آموزشی و آزمایشی استفاده شده است. این استراتژی اطمینان می‌دهد که هر دو کلاس به طور مناسب در هر بخش از داده‌ها نمایندگی شوند که باعث بهبود عملکرد مدل در تشخیص فیشینگ می‌شود.

۳-۶- پیاده‌سازی مدل‌های یادگیری گروهی و یادگیری ماشین

در این پژوهش، از شش مدل یادگیری گروهی و یادگیری ماشین برای مقایسه استفاده شده است:

- **ماشین بردار پشتیبان**: دقت بالا برای داده‌های خطی و غیرخطی.
- **درخت تصمیم**: سادگی و تفسیرپذیری بالا.
- **نزدیکترین همسایه**: قدرت تطبیق با الگوهای مختلف.
- **جنگل تصادفی**: کاهش بیش‌برازش و افزایش دقت.
- **روش تقویت مدل**: الگوریتمی قدرتمند برای داده‌های نامتوازن.
- **بهینه‌سازی مدل‌ها**: استفاده از GridSearchCV برای یافتن بهترین پارامترها، افزایش تعداد درخت‌ها در Random Forest و XGBoost، تنظیم C و γ در SVM برای افزایش دقت.

۳-۷- پیاده‌سازی مدل‌های یادگیری عمیق

سه مدل یادگیری عمیق برای بررسی عملکرد تشخیص فیشینگ استفاده شده است:

[1] Synthetic Minority Over-sampling Technique
[2] Adaptive Synthetic Sampling
[3] Information Gain
[4] Principal Component Analysis
[5] Stratified Sampling

- LSTM [1]: مناسب برای داده‌های ترتیبی
- GRU[2]: جایگزینی بهینه برایLSTM
- RNN [3]: شناسایی وابستگی‌های زمانی در داده‌ها

۳-۸- روش اعتبارسنجی متقابل

روش اعتبارسنجی متقابل[۴] یکی از مهم‌ترین روش‌های ارزیابی عملکرد مدل‌های یادگیری ماشین و یادگیری عمیق است که به طور مؤثری از پدیده بیش‌برازش[۵] جلوگیری می‌کند [۱۳]. در مسائلی مانند تشخیص وب‌سایت‌های فیشینگ که داده‌ها ممکن است شامل الگوهای پرتعداد یا ناهماهنگ باشند، اگر مدل تنها روی یک مجموعه آموزش و آزمون ثابت آموزش ببیند، ممکن است فقط روی همان داده‌ها عملکرد خوبی نشان دهد، ولی در داده‌های جدید و ناشناخته دچار افت کارایی شود. در روش اعتبارسنجی متقابل با تقسیم داده‌ها به چند بخش و استفاده متناوب از هر بخش برای آموزش و ارزیابی مدل، کمک می‌کند تا مدل به شکل کلی‌تری یاد بگیرد و وابسته به بخش خاصی از داده‌ها نباشد. در این مقاله، این رویکرد به‌ویژه برای الگوریتم‌های پیچیده‌تری چون LSTM، GRU و RNN که مستعد بیش‌برازش هستند، مفید است. با استفاده از این روش می‌توان اطمینان حاصل کرد که عملکرد مدل در کل مجموعه داده پایدار، قابل اعتماد و تعمیم‌پذیر است.

در مسائل دارای داده‌های محدود یا پرتنوع مانند تشخیص وب‌سایت‌های فیشینگ، استفاده از 10-Fold Cross-Validation توصیه می‌شود. زیرا تعادل بهتری بین بایاس و واریانس فراهم می‌کند و هم نوسانات ناشی از داده‌های نامتوازن را کاهش می‌دهد [۱۴].

۳-۹- تفسیرپذیری مدل با استفاده از روش SHAP[۶]

در دنیای امنیت سایبری، به‌ویژه در تشخیص حملات فیشینگ، صرفاً رسیدن به دقت بالا کافی نیست؛ بلکه درک منطق تصمیم‌گیری مدل‌های یادگیری ماشین نیز از اهمیت بالایی برخوردار است [۱۵]. مدل‌های یادگیری عمیق مانند LSTM، GRU و RNN به دلیل ساختار پیچیده و غیرخطی خود، اغلب به عنوان "جعبه سیاه" شناخته می‌شوند، به این معنی که فرایند تصمیم‌گیری آن‌ها از ورودی به خروجی برای انسان به راحتی قابل درک نیست. برای رفع این چالش و افزایش شفافیت و اعتمادپذیری مدل‌های توسعه‌یافته در این پژوهش، از روش توضیحات افزایشی SHAP استفاده شده است [۱۶]. با استفاده از این روش، به دنبال پاسخ به این پرسش هستیم: هر یک از ویژگی‌های ورودی (مانند طول URL، وجود کاراکتر "@" یا استفاده از پروتکل HTTPS) چه سهمی در تصمیم نهایی مدل برای طبقه‌بندی یک وب‌سایت به عنوان فیشینگ یا قانونی داشته است؟ SHAP با محاسبه سهم منصفانه هر ویژگی در فرآیند پیش‌بینی، به ما امکان می‌دهد تا تأثیر دقیق هر متغیر را بر خروجی مدل ارزیابی کنیم. این روش نه‌تنها تفسیرهای محلی برای هر پیش‌بینی منفرد ارائه می‌دهد، بلکه با تجمیع این تفسیرها، یک دید کلی از رفتار مدل و اهمیت ویژگی‌ها فراهم می‌آورد که این امر برای اعتبارسنجی و درک عمیق مدل حیاتی است. برای هر نمونه در مجموعه داده آزمایشی، مقادیر SHAP برای تمام ویژگی‌های ورودی محاسبه می‌شود. مقدار مثبت SHAP برای یک ویژگی نشان می‌دهد که آن ویژگی، پیش‌بینی مدل را به سمت کلاس "فیشینگ" سوق داده است، در حالی که مقدار منفی، آن را به سمت کلاس "قانونی" هدایت می‌کند. بزرگی این مقدار نیز نشان‌دهنده قدرت تأثیر آن ویژگی است.

۴- پیاده‌سازی و بررسی نتایج بدست‌آمده

در این بخش، مراحل پیاده‌سازی مدل‌های یادگیری گروهی، یادگیری ماشین و یادگیری عمیق برای تشخیص حملات فیشینگ بررسی شده و نتایج آزمایش‌ها تحلیل می‌شود. ابتدا محیط پیاده‌سازی و پارامترهای کلیدی مدل‌ها توضیح داده شده و سپس معیارهای ارزیابی معرفی می‌شوند. در نهایت، عملکرد مدل‌های مختلف از طریق مقایسه دقت، نرخ تشخیص و سایر معیارهای ارزیابی مورد بررسی قرار می‌گیرد .

۴-۱- محیط پیاده‌سازی[۷]

از موارد زیر برای شبیه‌سازی استفاده گردید:

- پلتفرم گوگل کولب[۸]
- زبان برنامه‌نویسی پایتون ۳.۱۱[۹]
- کتابخانه‌های مورد استفاده:
- Scikit-learn برای مدل‌های یادگیری ماشین
- TensorFlow/Keras برای مدل‌های یادگیری عمیق
- XGBoost برای الگوریتم تقویت‌شده گرادیان
- Imbalanced-learn برای متعادل‌سازی داده‌ها
- Matplotlib و Seaborn برای مصورسازی داده‌ها و نتایج

برای پیاده‌سازی روش SHAP بر روی مدل‌های LSTM، GRU و RNN، از ابزار *DeepExplainer* بهره گرفته شد که به طور خاص برای شبکه‌های عصبی عمیق بهینه‌سازی شده است.

۴-۲- انتخاب ویژگی و کاهش ابعاد

در این پژوهش، برای افزایش دقت مدل‌ها و کاهش پیچیدگی محاسباتی، از سه روش مختلف جهت تحلیل اهمیت ویژگی‌ها استفاده شده است:

[1] Long Short-Term Memory
[2] Gated Recurrent Unit
[3] Recurrent Neural Network
[4] Cross-Validation
[5] overfitting
[6] SHapley Additive exPlanations
[7] Implementation Environment
[8] Google Colab
[9] Python 3.11

اطلاعات متقابل: این روش با اندازه‌گیری میزان کاهش عدم قطعیت (آنتروپی) در صورت دانستن مقدار یک ویژگی خاص، اهمیت آن ویژگی را مشخص می‌کند که در شکل (۲) نشان داده شده است.

نسبت اطلاعات[1]: این معیار، مقدار اطلاعات متقابل را نسبت به آنتروپی ویژگی نرمال‌سازی کرده و تأثیر ویژگی‌هایی که تعداد مقادیر یکتای زیادی دارند را کاهش می‌دهد که در شکل (۳) آورده شده است.

تحلیل مؤلفه‌های اصلی: این روش کاهش ابعاد، با تبدیل داده‌ها به فضای جدیدی از مؤلفه‌های خطی مستقل، سعی دارد بیشترین واریانس داده را در کمترین تعداد مؤلفه حفظ کند که در شکل (۴) نشان داده شده است.

بر اساس این سه معیار، ویژگی‌های دارای اهمیت بیشتر انتخاب شده و ترکیبی از آن‌ها به همراه مؤلفه‌های اصلی استخراج‌شده توسط PCA، به عنوان ورودی مدل‌ها در نظر گرفته شده است. نمودارهای مربوط به اهمیت ویژگی‌ها نیز در بخش نتایج ارائه شده‌اند.

۴-۳- تنظیمات مدل‌ها و بهینه‌سازی پارامترها

یکی از الزامات اصلی در پژوهش‌های یادگیری ماشین و یادگیری عمیق، قابلیت بازتولید نتایج است. ازاین‌رو، در این پژوهش تمامی تنظیمات مدل‌ها (معماری، فراپارامترها، روش آموزش و محیط اجرا) به‌صورت دقیق گزارش می‌شود تا سایر پژوهشگران بتوانند نتایج را بازتولید و مقایسه کنند. در این مقاله، علاوه بر گزارش نتایج تک‌مرحله‌ای، برای کاهش اثر تصادفی بودن تقسیم داده‌ها و جلوگیری از بیش‌برازش، از اعتبارسنجی متقابل ۱۰ Fold-نیز استفاده شده است.

مدل‌های یادگیری ماشین:

- مدل ماشین بردار پشتیبان: استفاده از هسته RBF با C=100,gamma=scale.
- درخت تصمیم: تنظیم عمق درخت روی ۵۰.
- مدل نزدیکترین همسایه: انتخاب ۳ همسایه با وزن‌دهی.

مدل‌های یادگیری ماشین با پارامترهای نهایی زیر آموزش داده شدند (جدول ۴). پارامترها مطابق کد پیاده‌سازی نهایی انتخاب شده‌اند.

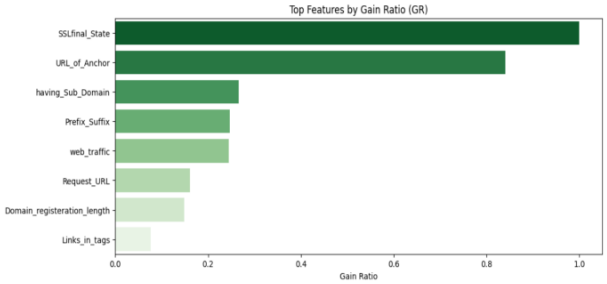
تنظیمات مدل‌های یادگیری گروهی/تقویتی(Ensemble) :

به‌منظور مقایسه منصفانه و دقیق، مدل‌های گروهی با پارامترهای نهایی زیر تنظیم شدند.

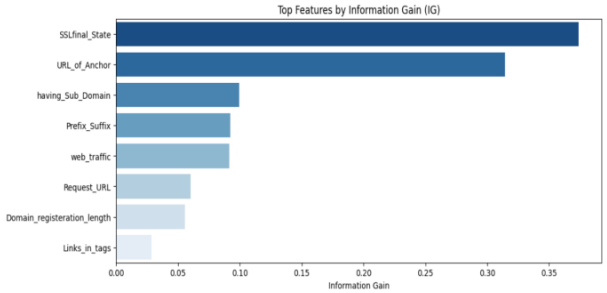
- جنگل تصادفی: استفاده از ۷۰۰ درخت تصمیم و عمق ۵۵.
- XGBoost: تنظیم تعداد ۶۰۰ درخت تصمیم با نرخ یادگیری ۰.۰۱.
- AdaBoost: تنظیم ۴۰۰ درخت ضعیف با نرخ یادگیری ۰.۸.

معماری و تنظیمات مدل‌های یادگیری عمیق(DL)

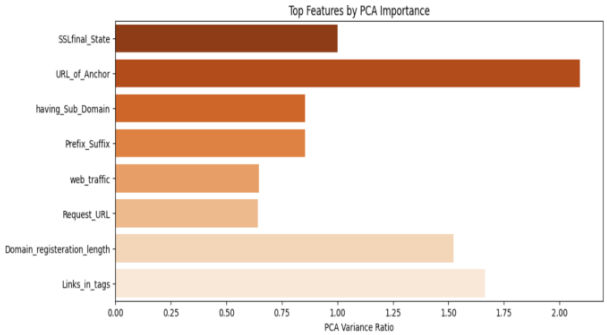
در این پژوهش سه مدل یادگیری عمیق LSTM، GRU و RNN پیاده‌سازی و مقایسه شدند. برای افزایش عملکرد و کاهش عدم‌پایداری آموزش، از یک ساختار مشترک اولیه مبتنی بر لایه‌های Dense به‌همراه نرمال‌سازی و Dropout استفاده شد و سپس لایه‌های بازگشتی (RNN/LSTM/GRU) روی این بازنمایی اعمال گردید.



شکل (۲): اندازه‌گیری مقدار اطلاعات متقابل ویژگی‌ها.



شکل (۳): اندازه‌گیری میزان کاهش عدم قطعیت ویژگی‌ها.



شکل (۴): اندازه‌گیری تحلیل موئلفه‌های اصلی ویژگی‌ها

جدول (۴): پارامترهای نهایی مدل‌های یادگیری ماشین

مدل	تنظیمات/فراپارامترهای نهایی
SVM	kernel = RB ،C = 100 gamma = scale ، probability = True ،random_state = 42
Decision Tree	max_depth = 50 min_samples_split = 5 random_state = 42
KNN	n_neighbors = 3 ، weights = distance

جدول (۵): پارامترهای نهایی مدل‌های یادگیری گروهی

مدل	تنظیمات/فراپارامترهای نهایی
Random Forest	n_estimators = 700 max_depth = 55 min_samples_split = 2 max_features = sqrt random_state = 42
XGBoost	n_estimators = 600 learning_rate = 0.01 max_depth = 25 random_state = 42 eval_metric = logloss
AdaBoost	n_estimators = 400 learning_rate = 0.8 random_state = 42

[1] Gain Ratio

جدول (۶): پارامترهای نهایی مدل‌های یادگیری عمیق

مدل	معماری
LSTM	Dense(2000, ReLU) → BN → Dropout(0.02) → Dense(1000, ReLU) → BN → Dropout(0.02) → LSTM(1000, ReLU, return_sequences=True) → LSTM(500, ReLU) → Dropout(0.02) → Dense(1, Sigmoid)
GRU	Dense(2000, ReLU) → BN → Dropout(0.02) → Dense(1000, ReLU) → BN → Dropout(0.02) → GRU(1000, ReLU, return_sequences=True) → GRU(500, ReLU) → Dropout(0.02) → Dense(1, Sigmoid)
RNN	Dense(2000, ReLU) → BN → Dropout(0.02) → Dense(1000, ReLU) → BN → Dropout(0.02) → SimpleRNN(2000, ReLU, return_sequences=True) → SimpleRNN(1000, ReLU) → Dropout(0.02) → Dense(1, Sigmoid)

جدول (۷): تنظیمات آموزش مدل‌های یادگیری عمیق

مولفه	مقدار
Optimizer	Adam
Learning Rate	0.000001 (1e-6)
Loss Function	Binary Cross-Entropy
Metric	Accuracy
Epochs	800
Batch Size	512
Verbose	0

بخش مشترک ابتدای شبکه[1]: برای هر سه مدلDL ، دو لایه Denseبه‌صورت زیر به‌کار رفت:

- Dense(2000) با فعال‌ساز ReLU
- Batch Normalization
- Dropout با نرخ ۰.۰۲
- Dense(1000) با فعال‌ساز ReLU
- Batch Normalization
- Dropout با نرخ ۰.۰۲

سپس بسته به نوع مدل، لایه‌های بازگشتی اعمال شد.

تنظیمات آموزش مدل‌های یادگیری عمیق: برای آموزش مدل‌های DL، تنظیمات جدول (۷) به‌کار گرفته شد.

کاهش بعد و آماده‌سازی ورودی نهایی: به‌منظور کاهش پیچیدگی محاسباتی و افزایش پایداری مدل، از PCA با 10 مؤلفه اصلی استفاده شده است. همچنین ۸ ویژگی منتخب بر اساس IG/GR با ۱۰ مؤلفه PCAادغام شدند؛ بنابراین بردار ورودی نهایی شامل ۱۸**ویژگی** (۸ ویژگی منتخب + ۱۰ مؤلفه) در نظر گرفته شد.

کنترل تصادفی بودن و بازتولیدپذیری

- تقسیم داده‌ها به آموزش/آزمون با نسبت ۸۰ به ۲۰ و به صورت نمونه‌گیری طبقه‌بندی‌شده انجام شد.
- مقدار random_state = 42 برای مدل‌های دارای مؤلفه تصادفی تنظیم شد.(SVM/Tree/Ensemble)
- علاوه بر ارزیابی معمول، از اعتبارسنجی متقابل Fold-۱۰ برای گزارش پایداری عملکرد استفاده شد .
- محیط اجرا شامل Google Colab و Python 3.11 و کتابخانه‌های Scikit-learn، TensorFlow/Keras، XGBoost و Imbalanced-learn بوده است.

۴-۴- معیارهای ارزیابی مدل‌ها

برای ارزیابی مدل‌های پیشنهادی، از معیارهای استاندارد زیر استفاده شده است [۱۳]:

- **معیار دقت**[2]: نسبت تعداد پیش‌بینی‌های صحیح به کل پیش‌بینی‌ها
- **مساحت زیر منحنی**[3]: قابلیت مدل در تشخیص نمونه‌های فیشینگ
- **دقت پیش‌بینی مثبت**[4]: نسبت پیش‌بینی‌های مثبت درست به کل پیش‌بینی‌های مثبت

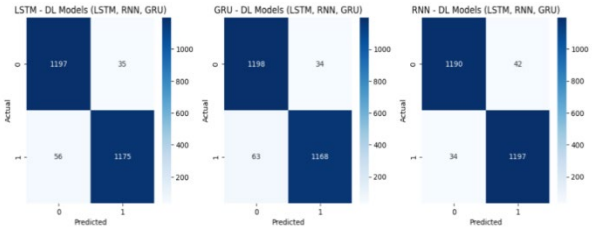
نرخ بازخوانی[5]: نسبت نمونه‌های فیشینگ که درست شناسایی شده‌اند.

امتیاز F1[6] : میانگین هارمونیک Precision وRecall

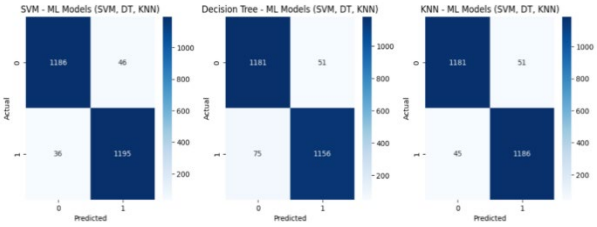
٤-٥- تحلیل عملکرد با استفاده از ماتریس درهم‌ریختگی

در ادامه برای تحلیل دقیق‌تر عملکرد مدل‌ها، از ماتریس درهم‌ریختگی[7] استفاده شده است. در شکل (۵-۷) این ماتریس‌ها برای هر گروه از مدل‌ها شامل الگوریتم‌های یادگیری عمیق، یادگیری ماشین و یادگیری گروهی رسم شده‌اند. ماتریس درهم‌ریختگی به‌طور خاص نشان می‌دهد که مدل تا چه اندازه قادر بوده است وب‌سایت‌های فیشینگ و غیرفیشینگ را به درستی تشخیص دهد. مقادیر TP، TN، FP و FN به‌صورت عددی قابل مشاهده‌اند و امکان محاسبه دقیق معیارهای مکمل مانند دقت Precision، Recall و F1 را فراهم می‌سازد. نتایج نشان می‌دهند که مدل RNN توانسته است تعداد خطاهای نوع دوم False Negative را به حداقل برساند که از اهمیت زیادی در کاربردهای امنیتی برخوردار است.

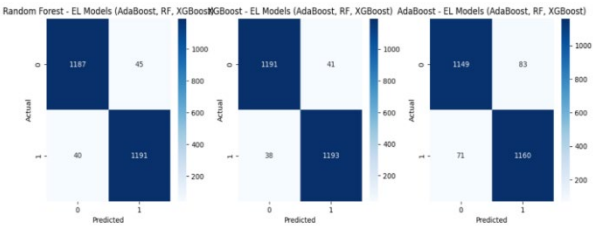
[1] Feature Projection Block
[2] Accuracy
[3] AUC-ROC
[4] Precision
[5] Recall
[6] F1-Score
[7] Confusion Matrix



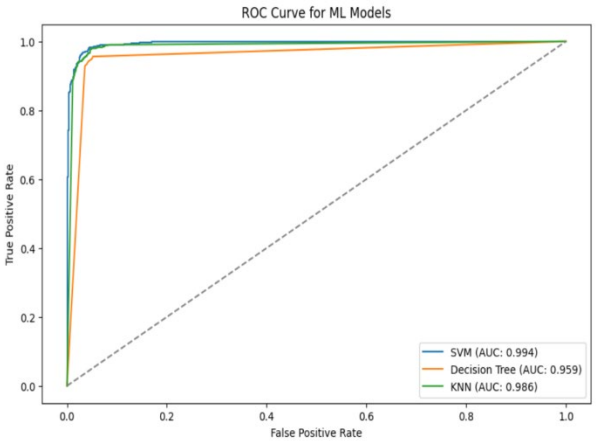
شکل (۵): ماتریس در هم ریختگی مدلهای یادگیری عمیق



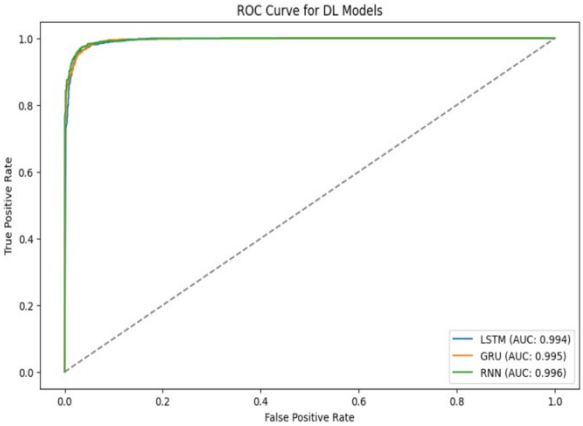
شکل (۶): ماتریس در هم ریختگی مدلهای یادگیری ماشین



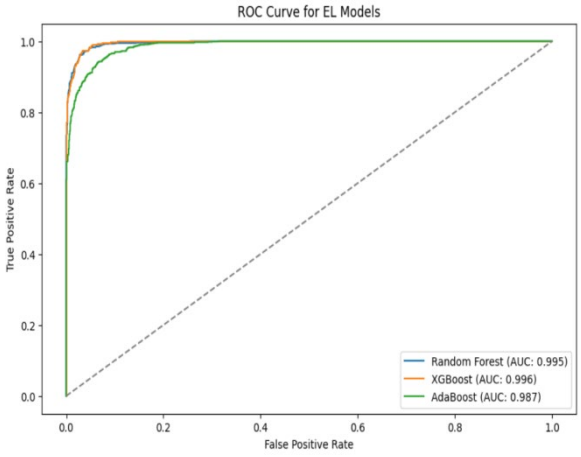
شکل (۷): ماتریس در هم ریختگی مدلهای یادگیری گروهی



شکل (۸): مساحت زیر نمودار مدلهای یادگیری ماشین



شکل (۹): مساحت زیر نمودار مدلهای یادگیری عمیق



شکل (۱۰): مساحت زیر نمودار مدلهای یادگیری گروهی

٤-٦- تحلیل عملکرد با منحنی مشخصه عملکرد گیرنده[1]

به‌منظور ارزیابی دقیق‌تر عملکرد مدل‌ها، از منحنی مشخصه عملکرد گیرنده نیز استفاده شده است. این نمودار با ترسیم نرخ مثبت واقعی[2] در برابر نرخ مثبت کاذب[3] برای مدل‌های مختلف، نمایی بصری از قدرت تفکیک مدل در تشخیص وب‌سایت‌های فیشینگ ارائه می‌دهد. هرچه مساحت زیر نمودار[4] به ۱ نزدیک‌تر باشد، مدل عملکرد بهتری دارد. نتایج نشان می‌دهد که مدل‌های پیشنهادی به‌ویژه مدل RNN و XGBoost دارای مقدار AUC بالا بوده و توانایی بالایی در تفکیک کلاس‌های فیشینگ و غیرفیشینگ دارند. نمودارهای ROC برای مدل‌های یادگیری ماشین، یادگیری گروهی و یادگیری عمیق در شکل‌های (۸-۱۰) نمایش داده شده‌اند.

٤-٧- تحلیل نتایج مدل‌ها

٤-٧-١- مقایسه و تحلیل عملکرد مدل‌های یادگیری گروهی، یادگیری ماشین و یادگیری عمیق

از جدول (۸) دیده می‌شود که روش RNN به دقت ۹۹٪ دست یافته است که نشان‌دهنده اثربخشی روش پیشنهادی است. XGBoost و Random Forest عملکرد مطلوبی دارند که به دلیل استفاده از الگوریتم‌های مبتنی بر درخت تصمیم است. SVM عملکرد قابل قبولی دارد اما برای داده‌های نامتوازن نیازمند تنظیمات بیشتر است. Decision Tree و AdaBoost عملکرد ضعیف‌تری نسبت به سایر مدل‌ها دارند. در جدول (۹) نتایج الگوریتم‌های پیشنهادی با استفاده از روش 10-fold Cross-Validation به صورت میانگین و انحراف معیار، برای معیارهای ارزیابی مدل ها محاسبه شده است. این روش به دلیل بررسی پایداری مدل روی ۱۰ تقسیم مختلف داده، از اعتبار بیشتری نسبت به نتایج تک‌باره برخوردار است.

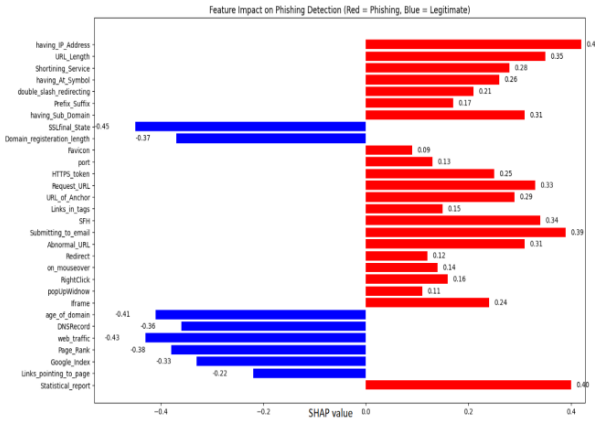
[1] ROC Curve
[2] True Positive Rate
[3] False Positive Rate
[4] AUC

جدول (۸): مقایسه عملکرد مدلهای مختلف

مدل	Accuracy	AUC-ROC	F1-Score	Precision	Recall
RNN	**99.01%**	**99.89%**	**99.01%**	**98.95%**	**99.07%**
XGBoost	96.83%	99.59%	96.82%	96.90%	96.75%
SVM	96.67%	99.44%	96.68%	96.29%	97.07%
LSTM	96.58%	99.56%	96.59%	96.51%	96.66%
Random Forest	96.50%	99.53%	96.52%	96.21%	96.83%
GRU	96.46%	99.50%	96.46%	96.50%	96.42%
KNN	96.10%	98.64%	96.11%	95.88%	96.34%
Decision Tree	94.96%	95.84%	94.91%	95.85%	93.98%
AdaBoost	93.54%	98.71%	93.58%	92.94%	94.23%

جدول (۹): مقایسه عملکرد مدلهای مختلف

مدل	Accuracy	AUC-ROC	F1-Score	Precision	Recall
RNN	**98.74% ± 0.32%**	**99.78% ± 0.11%**	**98.75% ± 0.34%**	**98.68% ± 0.37%**	**98.82% ± 0.31%**
LSTM	96.21% ± 0.45%	99.42% ± 0.18%	96.23% ± 0.47%	96.15% ± 0.50%	96.31% ± 0.43%
GRU	96.09% ± 0.48%	99.36% ± 0.20%	96.10% ± 0.50%	96.14% ± 0.52%	96.05% ± 0.46%



شکل (۱۱): مقادیر SHAP برای بررسی تاثیر ویژگی‌های مختلف

روش 10-fold با میانگین‌گیری روی ۱۰ تقسیم مختلف، پایداری مدل را دقیق‌تر ارزیابی می‌کند و نتایج محافظه‌کارانه‌تری ارائه می‌دهد.

٤-۷-۲- تفسیرپذیری مدل پیشنهادی با استفاده از روش SHAP

برای ارزیابی اهمیت ویژگی‌های ورودی در عملکرد مدل پیشنهادی، مقادیر SHAP برای هر ویژگی محاسبه شده‌اندکه در شکل (۱۱) نمایش داده شده است. تحلیل این مقادیر نشان می‌دهد که برخی ویژگی‌ها مانند استفاده از IP به‌جای نام دامنه[1]، طول URL[2]، استفاده از سرویس‌های کوتاه‌کننده[3] و استفاده از فرم‌های ارسال به ایمیل[4] بیشترین تأثیر مثبت را در پیش‌بینی فیشینگ بودن وب‌سایت‌ها داشته‌اند. در مقابل، ویژگی‌هایی مانند وجود گواهینامه SSL معتبر[5]، طول عمر ثبت دامنه[6] و میزان ترافیک وب‌سایت[7] بیشتر به سمت پیش‌بینی قانونی بودن سایت گرایش داشته‌اند.

٤-۸- مقایسه مدل‌های یادگیری ماشین و یادگیری عمیق

مزایای مدل‌های یادگیری ماشین: سرعت پردازش بالاتر و پیچیدگی کمتر، قابلیت تفسیرپذیری بیشتر (به‌ویژه در مدل‌های درختی)

مزایای مدل‌های یادگیری عمیق: دقت بالاتر در شناسایی الگوهای پیچیده، توانایی پردازش داده‌های ترتیبی و استخراج ویژگی‌های مفید

٥- نتیجه‌گیری و پیشنهادات برای کارهای آتی

در این پژوهش یک چارچوب ترکیبی برای تشخیص وب‌سایت‌های فیشینگ ارائه شد که زنجیره‌ای منسجم از پیش‌پردازش داده، متعادل‌سازی کلاس‌ها، انتخاب ویژگی و کاهش بُعد را پیش از مدل‌سازی به‌کار می‌گیرد. نتایج نشان داد که مدل RNN بهینه‌شده، بالاترین عملکرد را در میان مدل‌های بررسی‌شده ارائه می‌دهد و در کنار معیار دقت، از نظر کاهش خطاهای امنیتی به‌ویژه FalseNegative اهمیت ویژه‌ای دارد؛ زیرا در کاربردهای امنیت سایبری، خطای FN به معنای «عبور یک وب‌سایت فیشینگ به‌عنوان قانونی» است که پیامد عملی آن می‌تواند سرقت اطلاعات و خسارت مستقیم باشد .

از منظر تحلیلی، چند عامل در بهبود عملکرد مدل نقش کلیدی داشت:

متعادل‌سازی داده‌ها موجب شد مدل‌ها دچار سوگیری به سمت کلاس اکثریت نشوند و حساسیت[8] در تشخیص فیشینگ تقویت گردد. گزارش توزیع کلاس قبل/بعد از بالانس‌سازی نشان می‌دهد که این مرحله یکی از پایه‌های پایداری مدل است .

انتخاب ویژگی‌با معیارهای اطلاعاتی باعث حذف ویژگی‌های کم‌اثر و تمرکز مدل روی ویژگی‌های تعیین‌کنند (مانند SSLfinal_State و URL_of_Anchor) شد و در نتیجه هم دقت و هم تفسیرپذیری بهبود یافت.

کاهش بُعد با PCA با حذف هم‌بستگی و فشرده‌سازی اطلاعات در مؤلفه‌های اصلی، پیچیدگی محاسباتی را کاهش داد و زمینه را برای یادگیری پایدارتر در مدل‌های عمیق فراهم کرد .

در نهایت، به‌کارگیری **SHAP** به‌عنوان لایه تفسیرپذیری، «چرایی تصمیم مدل» را آشکار کرد و نشان داد کدام ویژگی‌ها بیشترین سهم

[1] having_IP_Address
[2] URL_Length
[3] Shortining_Service
[4] Submitting_to_email
[5] SSLfinal_State
[6] Domain_registeration_length
[7] web_traffic
[8] Recall

را در پیش‌بینی فیشینگ یا قانونی بودن دارند؛ این موضوع برای استفاده عملی در امنیت سایبری ضروری است.

مهم‌ترین نتایج این پژوهش به شرح زیر است:

- مدل RNN توانست به دقت ۹۹٪ دست یابد که نشان‌دهنده اثربخشی روش پیشنهادی در تشخیص سایت‌های فیشینگ است.
- مدل‌های مبتنی بر یادگیری عمیق عملکرد بهتری نسبت به مدل‌های یادگیری ماشین و یادگیری گروهی نشان دادند.
- روش‌های متعادل‌سازی داده‌ها SMOTE و ADASYN تأثیر چشمگیری بر افزایش دقت مدل‌ها داشتند.
- استفاده از روش‌های انتخاب ویژگی[1] باعث بهبود عملکرد مدل‌ها شد و ویژگی‌های غیرضروری حذف شدند.

۵-۱- چالش‌ها و محدودیت‌ها[2]

پردازش زمان‌بر در مدل‌های یادگیری عمیق: مدل‌های RNN و LSTM به دلیل داشتن ساختار پیچیده و وابستگی زمانی نیاز به منابع پردازشی قوی و زمان آموزش طولانی‌تر دارند.

عدم تفسیرپذیری بالا در مدل‌های یادگیری عمیق: مدل‌های یادگیری ماشین مانند Random Forest و XGBoost تفسیرپذیری بالایی دارند، اما مدل‌های یادگیری عمیق مانند RNN و LSTM رفتار پیچیده‌ای دارند و تحلیل خروجی آن‌ها دشوارتر است.

امکان نیاز به داده‌های بیشتر: اگرچه مدل‌های ما دقت بالایی داشتند، اما ممکن است با داده‌های حجیم‌تر و واقعی‌تر عملکرد متفاوتی داشته باشند.

۵-۲- پیشنهادات برای کارهای آتی[3]

پیشنهادات برای بهبود روش پیشنهادی و تحقیقات آینده:

ترکیب یادگیری ماشین و یادگیری عمیق[4]: استفاده از ترکیب مدل‌های یادگیری ماشین و یادگیری عمیق مانند ترکیب XGBoost با RNN برای بهره‌گیری از مزایای هر دو روش.

استفاده از شبکه‌های عصبی کانولوشنی[5] در کنار RNN برای بهبود عملکرد مدل در تشخیص ویژگی‌های مهم در URL ها.

استفاده از یادگیری تقویتی[6]: بهره‌گیری از الگوریتم‌های یادگیری تقویتی برای شناسایی و مقابله هوشمندانه‌تر با حملات فیشینگ.

توسعه مدل‌های سبک‌تر و بهینه‌تر: بهینه‌سازی مدل‌ها برای کاهش زمان پردازش و مصرف منابع محاسباتی، مخصوصاً در مدل‌های یادگیری عمیق استفاده از روش‌های کاهش ابعاد و انتخاب ویژگی‌های مهم‌تر برای سبک‌تر کردن مدل‌ها.

استفاده از داده‌های گسترده‌تر: گسترش مجموعه داده‌ها با استفاده از داده‌های واقعی و به‌روز از وب‌سایت‌های فیشینگ.

استفاده از وب‌اسکریپینگ[7] برای جمع‌آوری داده‌های معتبر از وب.

ترکیب روش‌های امنیت سایبری با مدل‌های یادگیری: ادغام روش‌های مبتنی بر تحلیل ترافیک شبکه و الگوریتم‌های پردازش زبان طبیعی[8] برای تشخیص فیشینگ در ایمیل‌ها و پیام‌های متنی.

پیاده‌سازی مدل‌ها در سیستم‌های واقعی و عملیاتی: توسعه و پیاده‌سازی یک سیستم تشخیص فیشینگ مبتنی بر API که بتواند به‌صورت بلادرنگ[9] سایت‌های فیشینگ را شناسایی کند.

ایجاد یک افزونه مرورگر[10] برای تشخیص و هشدار درباره سایت‌های فیشینگ.

ترکیب CNN با RNN (CNN–BiRNN): برای استخراج بهتر الگوهای ساختاری از URL/HTML و سپس مدل‌سازی وابستگی‌ها با RNN، که می‌تواند هم دقت را افزایش دهد و هم خطای FN را کاهش دهد.

استفاده از داده‌های واقعی و به‌روز و ارزیابی عملیاتی: آزمون مدل روی داده‌های جدید (مانند PhishTank یا داده‌های جمع‌آوری‌شده با وب‌اسکریپینگ) و بررسی در سناریوهای واقعی.

پیاده‌سازی بلادرنگ و توسعه ابزار کاربردی :طراحی یک API عملیاتی یا افزونه مرورگر که بتواند در زمان واقعی هشدار دهد و علاوه بر خروجی مدل، توضیح SHAP را نیز ارائه کند.

افزودن لایه‌های امنیت داده (مانند بلاک‌چین یا ثبت تغییرات): برای افزایش اعتماد در تبادل داده و ثبت رخدادهای حمله/دفاع، به‌ویژه در محیط‌های سازمانی.

تحلیل پایداری در برابر حملات خصمانه(Adversarial): بررسی اینکه آیا مهاجم می‌تولند با تغییرات کوچک در ویژگی‌ها مدل را فریب دهد یا خیر، و طراحی راهکارهای مقاوم‌سازی.

۵-۳- جمع‌بندی نهایی

در این پژوهش، مدل‌های یادگیری گروهی، یادگیری ماشین و یادگیری عمیق برای شناسایی سایت‌های فیشینگ بررسی شدند و با استفاده از روش‌های متعادل‌سازی داده‌ها و انتخاب ویژگی، عملکرد مدل‌ها بهینه شد. نتایج نشان داد که مدل پیشنهادی RNN (۹۹٪) بهترین عملکرد را دارد، در حالی که مدل‌های XGBoost و Random Forest نیز دقت بالایی ارائه کردند. پیشنهاد شد که در آینده از روش‌های ترکیبی، یادگیری تقویتی، پردازش ترافیک شبکه و توسعه سیستم‌های عملیاتی برای بهبود بیشتر استفاده شود. نتایج این پژوهش می‌تواند به بهبود امنیت سایبری و جلوگیری از حملات فیشینگ در سطح گسترده کمک کند.

[1] Feature Selection
[2] Challenges and Limitations
[3] Future Work
[4] Hybrid Models
[5] CNN
[6] Reinforcement Learning
[7] Web Scraping
[8] NLP
[9] Real-time
[10] Browser Extension

مراجع

[1] Anti-Phishing Working Group (APWG). (2022). Phishing Activity Trends Report. https://apwg.org/trendsreports/

[2] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, 321–357. https://doi.org/10.1613/jair.953

[3] Mohammad, R. M., Thabtah, F., & McCluskey, L. (2015). Intelligent phishing detection system based on features extraction and analysis. *Information Security Journal: A Global Perspective*, 24(1-3), 123–134. https://doi.org/10.1080/19393555.2015.1051670

[4] Alnemari, S., & Alshammari, M. (2023). Detecting Phishing Domains Using Machine Learning. *Applied Sciences*, 13(8), 4649. https://doi.org/10.3390/app13084649

[5] Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications*, 117, 345–357. https://doi.org/10.1016/j.eswa.2018.09.029

[6] Zhang, Y., Ruan, X., Wang, H., & He, S. (2017). Twitter Trends Manipulation: A First Look Inside the Security of Twitter Trending. *IEEE Transactions on Information Forensics and Security*, 12(1), 144–156. https://doi.org/10.1109/TIFS.2016.2604226

[7] Catal, C., Giray, G., Tekinerdogan, B., & Kumar, S. (2022). Applications of deep learning for phishing detection: A systematic literature review. *Knowledge and Information Systems*, 64, 1457–1500. https://doi.org/10.1007/s10115-022-01672-x

[8] Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*,40(1),16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024

[9] Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

[10] Zhang, Y., Hong, J., & Cranor, L. (2007). CANTINA: A content-based approach to detecting phishing websites. *Proceedings of the 16th international conference on World Wide Web*, 639–648. https://doi.org/10.1145/1242572.1242659

[11] Verma, R., & Das, A. (2017). What's in a URL: Fast Feature Extraction and Malicious URL Detection. *Proceedings of the 3rd ACM on International Workshop on Security And Privacy Analytics*, 55–63. https://doi.org/10.1145/3041008.3041016

[12] Gorriz, J. M., Clemente, R. M., Segovia, F., Ramirez, J., Ortiz, A & ,Suckling, J. (2024). Is K-fold cross validation the best model selection method for Machine Learning ?. *arXiv preprint arXiv:2401.16407* .doi: 10.48550/arXiv.2401.16407.

[13] Mahlich, C., Vente, T & ,Beel, J. (2024). From Theory to Practice: Implementing and Evaluating e-Fold Cross-Validation. In *Proceedings of the International Conference on Artificial Intelligence and Machine Learning Research (CAIMLR* .(doi: 10.5281/zenodo.10648834.

[14] Jain, P & ,Verma, D. (2024). Enhancing Phishing Detection through Feature Importance Analysis and Explainable AI: A Comparative Study of CatBoost, XGBoost, and EBM Models .*arXiv preprint arXiv:2411.06860* .doi: 10.48550/arXiv.2411.06860

[15] Mishra, R & ,Kumar, S. (2024). XAI-PhD: Fortifying Trust of Phishing URL Detection Empowered by Shapley Additive Explanations .*arXiv preprint arXiv:2407.12648* . doi: 10.48550/arXiv.2407.12648.

[16] Alshingiti, Z.; Alaqel, R.;Al-Muhtadi, J.; Haq, Q.E.U.; Saleem,K.; Faheem, M.H. A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN. Electronics 2023, 12, 232. https://doi.org/10.3390/electronics12010232

[17] Kunndra, C., Choudhary, A., Kaur, J., Jogia, A., Mathur, P., & Shukla, V. (2023, October). NTPhish: A CNN-RNN Hybrid Deep Learning Model to Detect Phishing Websites. In *International Conference on Cryptology & Network Security with Machine Learning* (pp. 587-599). Singapore: Springer Nature Singapore.

[18] Kehkashan, T., Abdelhaq, M., Al-Shamayleh, A.S. et al. Explainable phishing website detection for secure and sustainable cyber infrastructure. Sci Rep 15, 41751 (2025). https://doi.org/10.1038/s41598-025-27984-w