Developing A Real-time Phishing Detection Chrome Extension Using Support Vector Machine (SVM) With Splunk Integration

Mohammed Ahmed Ali Abdualrhman [1]
Yousra M. Othman [2]
Nosiba R. Ahmed [2]
Nail Adeeb Ali Abdu [1]

Received: ٥٣/١١/2025
Revised: ٠٩/١١/2025
Accepted: 16/12/2025

[1] Information Technology Department, Faculty of Engineering & Computing, University of Science and Technology, Aden, Yemen.
[2]Cybersecurity Department, Faculty of Engineering & Computing, University of Science and Technology, Aden, Yemen.
*Corresponding Author's Email: moh_alherwi@hotmail.com

Developing A Real-time Phishing Detection Chrome Extension Using Support Vector Machine (SVM) With Splunk Integration

Mohammed Ahmed Ali Abdualrhman
Information Technology Department, Faculty of Engineering & Computing, University of Science and Technology,
Aden, Yemen.
Email: moh_alherwi@hotmail.com
M.alherwi@ust.edu

Yousra M. Othman
Cybersecurity Department, Faculty of Engineering & Computing, University of Science and Technology,
Aden, Yemen.
Email: sariossam770@gmail.com

Nosiba R. Ahmed
Cybersecurity Department, Faculty of Engineering & Computing, University of Science and Technology,
Aden, Yemen.
Email: 4securcyber@gmail.com

Nail Adeeb Ali Abdu
Information Technology Department, Faculty of Engineering & Computing, University of Science and Technology,
Aden, Yemen.
Email: n3a_alkershi@yahoo.com
n.a.alkershi@ust.edu

Abstract—Phishing attacks have become a widespread worldwide concern due to the negative impact on individuals and organizations by targeting valuable information, which may cause significant harm on various levels. Therefore, to mitigate these attacks, we proposed a system that uses a machine learning model to detect fraudulent URLs in real time through a Chrome extension and report them to Splunk, which will enable centralized analysis and supervision. The system utilizes a Support Vector Machine (SVM)—mainly used for text classification—with a Radial Basis Function (RBF) kernel. It was trained on a dataset of 11,054 records encompassing 30 unique features. The model demonstrated a robust and accurate performance in detecting phishing URLs, as it gained a score of 96.9% for accuracy, 96.5% for precision, 98% for recall, and 97.2% for F1 score. Ultimately, the primary purpose is to protect organizations and individuals from the severe consequences of phishing attacks in a real-time manner by detecting phishing URLs and then reporting them along with relevant information to Splunk.

Keywords— *Phishing Detection, Phishing URLs, Machine Learning, Real-time Detection, SVM Classifier, Splunk, Chrome Extension.*

I. INTRODUCTION

Throughout the rapid expansion in the digital world that has facilitated people's lives by providing several assistant online services, a dangerous side effect represented by cybercrime and attacks has occurred; at the top of the list are phishing attacks. Phishing fundamentally adopts the concept of deception, as it relies on techniques that help mimic trusted entities for the purpose of deceiving and tricking users to seize their personal information [1]. As it is one of the most common cybercrimes, over 298,000 complaints were reported by the Internet Crime Complaint Center [2] last year, accounting for roughly 34% of its caseload. In conjunction, APWG [3] (Anti-Phishing Working Group) observed 989,123 phishing attacks in the last quarter of 2024. In addition to e-mail, phishing has expanded to include modern communication technologies such as VoIP, social networking sites, instant messages, and online multiplayer games [4].

Since phishing attacks do not only target significant information, they are also used to spread other malware applications. This motivated us to obtain this research by developing a real-time Chrome-based extension that detects phishing websites and alerts users, with the reporting of visited phishing websites to Splunk for organization as an informing reporting system about employee activity.

Although anti-phishing website solutions do exist and are common, their efficiency is limited. Existing anti-phishing methodologies, such as the blacklist-based approach, heuristic detection, and AI- and ML-powered solutions, are limited in various aspects. The blacklisted approach requires exact matches to detect phishing websites. On the other hand, heuristic methods identify malicious payloads never seen before with higher probability but at the same time create a risk of misidentifying legitimate websites [5], while most ML approaches have shown promise in dynamically identifying phishing patterns based on features extracted from URLs and website content. However, many of these approaches operate offline or are not integrated into real-time user environments.

The primary aim of our research is developing a real-time phishing detection system implemented as a Google Chrome browser extension. The key idea of the system is to leverage a Support Vector Machine (SVM) classifier with a Radial Basis Function (RBF) kernel to analyze URLs in real time and determine whether they are malicious. The SVM classifier was trained on a dataset of 11,054 URLs, each characterized by 30 extracted features. When a phishing website is detected, the extension instantly alerts the user and

simultaneously logs the event into Splunk. This enables both individual protection and organizational monitoring, offering valuable insights for IT administrators overseeing employee browsing behavior and responding to security incidents.

The remainder of the paper is organized as follows. Section II discusses related works of the research. Section III describes the methodology, which includes general architecture. Section IV evaluates the effectiveness of the phishing detection system. Lastly, Section V outlines the conclusion of the paper and explores potential recommendations for future improvements.

II. Background & Related Works

The concept of utilizing a "bait" to induce a victim is referred to as "fishing," from which the term "phishing" is derived, following the same mechanism of deception to obtain valuable information [6]. Phishing primarily depends on social engineering techniques for exploitation. Authors of [6] discuss phishing technique's general schema that is derived into medium, vector, and technical approach, where these three components are interlinked together. The medium refers to the procedure that an attacker utilizes to transmit phishing attacks to the victim, such as the internet and short messaging service. Phishing vectors and technical approaches are extensively outlined later in this chapter.

A. URL Structure & Components

URL is a short form of Uniform resource locater to locate and retrieve a resource. It is composed of many parts [7], as illustrated in Fig. 1.

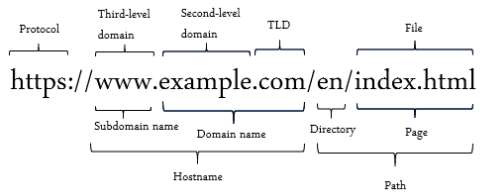

Fig. 1. URL Structure

As illustrated in figure 1, a URL mainly consists of a protocol, hostname, and path. The protocol is used to fetch the requested resources. The hostname, which is made up of many parts, identifies the host where the resource is located. The path is the directory on the web server.

The deep understanding of these components' manipulation techniques used by phishers helps in developing effective phishing detection systems and ensures sensitive information security.

B. Phishing Vectors and Technical Approaches

As stated above, the phishing techniques schema is composed of the medium, vector, and technical approaches. The vector is the method of the attack, as it forms what channel the attacker uses to conduct a phishing attack. Here are some common phishing vectors:

1. Vishing: Vishing, also called voice phishing, is where the attacker uses social engineering to deceive users into calling numbers and giving their financial information.

2. Smishing: Smishing, short for SMS phishing, is a scam in which scammers impersonate trusted entities such as banks and disseminate deceptive messages to divulge sensitive information or click on harmful links, leveraging social engineering techniques. [1].

3. Websites: Fake websites that allow attackers to exploit the login process and steal users' credentials by mimicking benign websites and utilizing multiple techniques [6].

4. Email Phishing: Fraudulent emails that mimic well-known organizations' emails to trick recipients into divulging private information or compromising systems by urging recipients to download malware or visit harmful links through the use of social engineering [8].

On the other hand, the technical approaches are the methodologies applied with social engineering and cross-site scripting during attacks.

1. Spear Phishing: Unlike email phishing, spear phishing specifically targets particular individuals or groups utilizing highly customized emails. Attackers would urge recipients to click on suspicious links or to install malicious files using social engineering techniques. [8].

2. Whale Phishing: Whaling is a targeted method of phishing in which the attacker aims to compromise a system by systematically targeting high-powered authority figures and prominent positions through prompting them to download malicious files [6].

3. Link Manipulation: Link manipulation refers to the mechanism by which an attacker spoofs a legitimate link in order to deceive users and lure them into his malicious website instead of the real one [9].

4. Clone Phishing: Clone phishing is a devious attack in which a legitimate email that has been sent with an attachment or a link is cloned and replaced with malicious links or attachments [1].

C. Related Works

Many studies have focused on leveraging machine learning approaches to perform the process of detecting phishing attacks, outlined below:

Leveraging two phishing datasets, the authors of [10] tested seven machine learning algorithms to obtain good classification results and to guarantee the classifier's normality. The voting classifier that combined RF, XGBoost, and MLP obtained the highest score with a 98.1%

accuracy result for the first dataset with 32 features selected, and for the second dataset, all classifiers obtained the same accuracy score with 50 features selected.

A novel approach for phishing URL detection was developed by A. Olayinka Taofeek [11]. This approach scrapes HTML code of the websites that are not in the database and then stores it as a string. Subsequently, an image creation module is used to convert the binary file into a 2D image via the BinVis method. This image is then analyzed using TensorFlow to perform classification. The model achieved an 82.6% accuracy score for the decision tree method, 83.4% for random forest, 86.3% for MLP, 86.4% for XGBoost, 81.8% for auto encoder, and 81.8% for SVM.

In [12], the authors used a balanced dataset that contains 11000 phishing and legitimate records. They got a sum of 17 features after applying a feature selection approach. Furthermore, they used a heuristic approach and ML for phishing URL detection. The XGBoost algorithm has scored an accuracy of 98.4%, which outperforms all other algorithms.

The authors of [13] analyzed URLs using mutual information for selecting features from a modern dataset with the use of logistic regression techniques. The proposed model has scored a high accuracy rate of 99.97% for 5 feature selections, 97.44% for 10, 96.63% for 15, 99.36% for 20, and 99.25% for 25, respectively.

Authors of [14] created a model for detecting phishing URLs from legitimate ones. A total of 15 features are taken into consideration. The model obtained 88.3% accuracy for the Random Forest classifier, 85% for Decision Tree, 87.8% for Logistic Regression, 87.1% for SVM, and 89.5% for LightGBM.

Researchers in [15] focused more on real scenarios of phishing website detection. Their methodology leveraged URL, HTML, and web technology features. The methodology does not depend on third-party services. The LightGBM classifier achieved 97.95% accuracy, and a complete set of 54 selected features got the best experimental results.

For [16], the authors proposed a novel approach using their own dataset. They trained an XGBoost classifier utilizing new features as URL character sequences without prior phishing knowledge, hyperlink information, and webpage texts. The model achieved an accuracy of 96.76% (1.39% FPR) on the custom dataset and 98.48% (2.09% FPR) on a benchmark dataset.

The researchers of [17] presented a real-time phishing website detection approach utilizing multiple machine learning algorithms using a hybrid set of URLs and hyperlink features. Accuracy values varied and gradually improved as logistic regression achieved 96.25%, followed by support vector machine (96.92%), decision tree (97.67%), random forest (98.75%), and XGBoost (99.17%).

Researchers of [18] proposed a comprehensive methodology for detecting phishing attacks, where they transformed URLs into fixed-size matrices via character embedding. They employed CNN models to extract features at contrastive levels and RF models to classify multi-level features. Finally, they presented the predicted outcomes using a winner-take-all approach. It's good to mention that web content and third-party services were not considered. The model achieved a 99.35% correct classification rate on their dataset and 99.26% on the benchmark data.

Authors of [19] developed a website that detects phishing websites depending on a machine learning approach. The dataset has 29 features and was trained on two classifiers; Decision Tree scored an accuracy of 95.54%, and Random Forest scored 97.31% accuracy.

III. Research Methodology

The proposed methodology as shown in Fig. 2. Includes data collection, feature engineering, model development and training using SVM classifier and model deployment, the chrome browser extension and the Splunk association process.

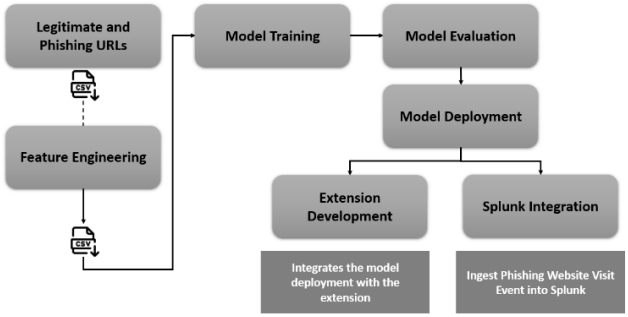

Fig. 2. Proposed Methodology

A. Dataset Description

The dataset [20] utilized in this work consists of 11054 URL samples, divided into 6157 as legitimate URLs and 4897 as phishing. It contains an Index column, 30 unique features, and a class column that indicates whether the URL is phishing or not.

B. Feature Engineering

Feature engineering is a crucial process that mainly involves feature extraction and data preprocessing.

1. Feature Extraction: Effective detection of phishing URLs relies heavily on the identification and extraction of relevant features. Based on the work of [32], thirty distinct features were selected and categorized into four primary groups as depicted in Table 1: Address Bar-based Features, Abnormal-based Features, HTML and JavaScript-based Features, and Domain-based Features. This step is done in a real-time manner, as features are extracted immediately after inputting the URL.

2. Data Preprocessing: Data preprocessing forms a significant phase that impact the model performance as crude data is not suitable to use and affect it negatively. It involves

employing many techniques, such as data cleaning, which consists of checking missing values, irregular data, duplicated data, and inconsistent data. Owing to the fact that the utilized dataset is already cleaned, this step was intentionally skipped.

Table 1. Features List and Categorization

Category	Features	Sum
Address Bar Based Features	1. Using IP 2. Long URL 3. Short URL 4. URL contains "@" Symbol 5. Redirecting "//" 6. Prefix or Suffix "-" 7. Sub Domain 8. HTTPS 9. Domain Registration Length 10. Favicon 11. Non-Standard Port 12. HTTPS in Domain	12
Abnormal Based Features	1. Request URL 2. Anchor URL 3. Links in Script Tags 4. Server Form Handler 5. Info Email 6. Abnormal URL	6
HTML and JavaScript Based Features	1. Status Bar Customization 2. Disabling Right Click 3. Using Pop-up Windows 4. Iframe Redirection 5. Website Forwarding	5
Domain Based Features	1. Age of Domain 2. DNS Record 3. Website Traffic 4. Page Rank 5. Google Index 6. Links Pointing to Page 7. Statistics Report	7

C. Model Training

Generally, machine learning algorithms' performance on new data is assessed using the train-test split approach. In this study, we utilized GridSearchCV for hyperparameter tuning due to its effectiveness and simplicity. A cross-validation technique was applied during the data-splitting process, which divides data into multiple folds. For each iteration, a fold is used as a validation set while the remaining folds compose the training set. This approach will prevent the model's overfitting and ensure reliable performance. We divided the dataset into 80% for training and 20% for testing. The training set is then applied with 5-fold cross validation, where four subsets are for training and the remaining one for validation. Moreover, the random state was set to 42 to maintain that shuffled data is consistently reproducible.

Support Vector Machine (SVM) is a highly powered supervised machine learning algorithm that gained its importance due to its ability to solve pattern recognition problems. The model works on labeled data to be able to classify new data into a predefined category [21].

Basically, the SVM structure is composed of four main components: the separating hyperplane, the maximum margin hyperplane, the soft margin, and the kernel function.

SVM has many kernels, yet two common kernels are linear and radial basis function (RBF). Typically, the RBF is usually a good first choice. It can handle nonlinear relationships between class labels and attributes as it maps samples into a higher-dimensional space. Moreover, the RBF kernel can perform as a linear kernel when configured with certain parameters, which distinguishes it as a more flexible choice. [22]. Furthermore, the model's RBF kernel parameters are illustrated merely below:

1. C (Regularization parameter), which is used to control the trade-off between low training error and low testing error. Lower values of C produce smoother decision boundaries resulting in stronger regularization, while higher values lead to sharper boundaries and weaker regularization. For hyperparameter tuning, we set the C value range between [0.1, 1, 10, 100], where 0.1 corresponds to high regularization and 100 corresponds to a lower one.

2. Gamma (Kernel Coefficient) defines the farness of a single training sample influence. Its low values result in a smooth decision boundary that may underfit data, and vice versa for higher values. For our model, we set its value range between [1, 0.1, 0.01, 0.001], which will define overfit and underfit for smooth and sharp boundaries.

Both parameters are evaluated using GridSearchCV for selecting the best combination.

D..Model Evaluation

In order to evaluate how effectively the model will discriminate between classes, the following metrics are used:

1. Accuracy: Percentage of correct predictions of phishing and legitimate websites out of the total number of websites [23].

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.1)$$

2. Precision: Percentage of correctly predicted as phishing out of all predicted phishing websites [23].

$$Precision = \frac{TP}{TP+FP} \quad (3.2)$$

3. Recall: Percentage of correctly predicted phishing websites out of all actual phishing websites [23].

$$Recall = \frac{TP}{TP+FN} \quad (3.3)$$

4. F1 Score: Harmonic values indicated precision and recall [23].

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (3.4)$$

5. Confusion Matrix: It is a N x N matrix, where N is the number of target classes. As illustrated in Fig. 3, its schema is constituted of True Positives (TP), False Positives (FP), False Negatives (FN) and True Negatives (TN). It identifies model's error through comparing predicted outputs to the actual ones.

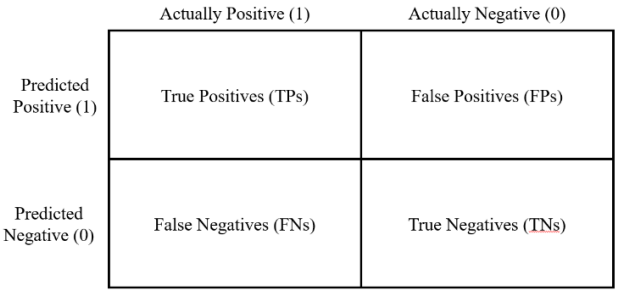

Fig. 3. Confusion Matrix

IV. RESULTS

A. Experimental Results

Upcoming outcomes are gained once our model is trained using the SVM classifier:

According to GridSearchCV, the optimal parameters were identified as C=10 and Gamma=0.1 using an RBF kernel. This combination achieved the highest average accuracy value of 96.49% for 5-fold cross-validation. Fig. 4 displays cross-validation scores for each fold on the training set, showing a range from 96.1% to 96.9%.

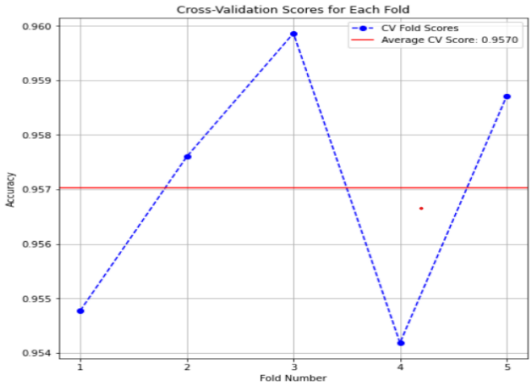

Fig. 4. Cross-Validation Score

Table 2. shows distinguishing of the four components of the confusion matrix; TP, FP, FN & TN.

Table 2. Confusion Matrix and its Components

Category	Confusion Matrix			
	TP	***FP***	***FN***	***TN***
Train	3839	82	50	4872
Test	932	44	25	1210

Fig. 5 indicates score values, respectively, for all employed metrics on testing sets. For the training set, our model achieved an accuracy of 98.5%, a precision of 98.3%, a recall of 99%, and an F1-score of 98.7%. On the other hand, the model has achieved an accuracy of 96.9%, a precision of 96.5%, a recall of 98%, and an F1-score of 97.2% on the testing test.

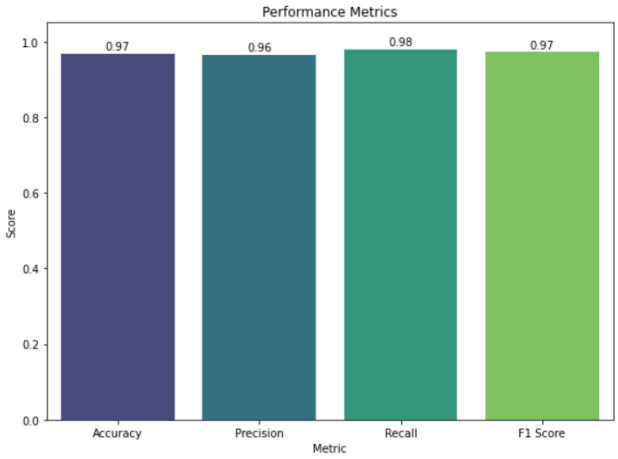

Fig. 5. Performance Metrics

As Fig. 6 shows, we calculated the inference latency taken by our model in order to get a prediction for 50 runs; the term "run" refers to one complete execution of a model inference, which will help in getting reliable latency statistics as latency differs for each run. In each run, we do calculate latency for the whole set of data.

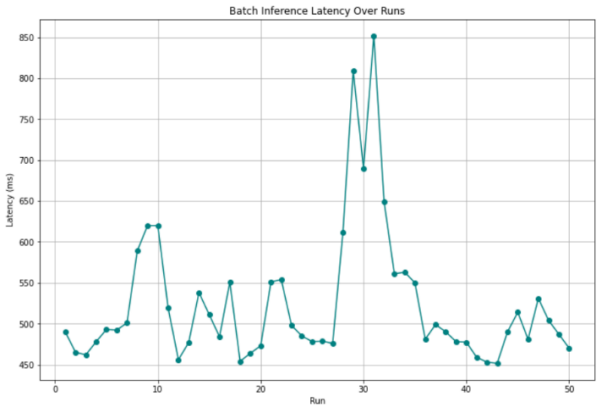

Fig. 6. Inference Latency

B. Comparison with Existing Works

We considered a total of five models that used different algorithms for the same purpose of this study; all used the same dataset, as Table 3 illustrates.

For the first three studies, which used different classifiers than SVM and scored 89.3%, 94.79%, and 95.6%, we can notice that the SVM classifier is more efficient for classifying binary classes, such as phishing URL detection.

For the fourth paper, which applied the feature selection technique, resulting in a sum of 19 features. Although it applied feature selection, our model scored better accuracy using the SVM model.

In the fifth study, accuracy achieved by utilizing different classifiers does not surpass accuracy achieved by the SVM classifier.

In the sixth one, they utilized another balanced dataset of the same size but applied a feature selection mechanism out of 48 features that have affected the model's performance.

Table 3. Comparison with Previous Works

Ref	Algorithm	Features	Key Metrics			
			Accuracy	*Precision*	*Recall*	*F1*
[24]	ELM	30	89.3%	91.5%	89.3%	90.4%
[25]	RF	30	94.79%	94.8%	94.8%	94.8%
[7]	Esnemble (bagging..)	30	95.4%	93.5%	95.5%	94.7%
[26]	RF	19	95.6%	97%	92.7%	95%
[27]	DNN with ADAM	30	96%	-	-	-
[28]	RF	10	96.17%	-	-	-
	SVM (proposed system)	30	96.9%	96.5%	98%	97.2%

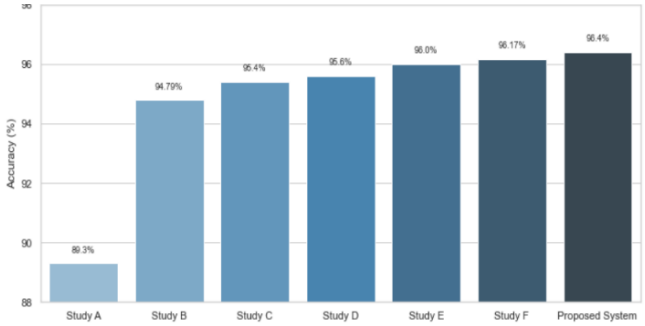

Fig. 7. Accuracy Comparison Among Studies

C. Model Deployment

Subsequent to the training and evaluation phases, the model is serialized using Python's pickle library, which saves it in a binary (.pkl) format. This allows the model to be loaded by a Flask server for real-time prediction without retraining.

D. Chrome Extension Development

A Chrome extension was developed to apply real-time detection of phishing URLs; as it captures URLs, it sends them in JSON format to the backend Flask server to extract features and classify them.

Fig. 8. Shows extension notification for legitimate URLs.

Fig. 8. Legitimate URL

Fig. 9. Shows extension notification for phishing URLs.

Fig. 9. Phishing URLs

To evaluate the model's performance after implementation, latency was measured over multiple requests executed through the Chrome extension.

Across a total of 15 requests, the system achieved an average model-prediction latency of 2.68 milliseconds.

However, the feature extraction process incurred significantly higher average latency of 19,848.208 milliseconds, resulting in a total average latency of 19,851.128 milliseconds. Because the feature extraction phase involves third-party features, it has clearly contributed substantially to the overall latency.

E. Splunk Integration

When a user visits a phishing URL, information such as the URL, IP address, hostname, message, and many others is ingested into Splunk using HEC over HTTP/HTTPS, as Fig. 10 illustrates. Furthermore, a report is created using Python SDK, as Fig. 11 depicts.

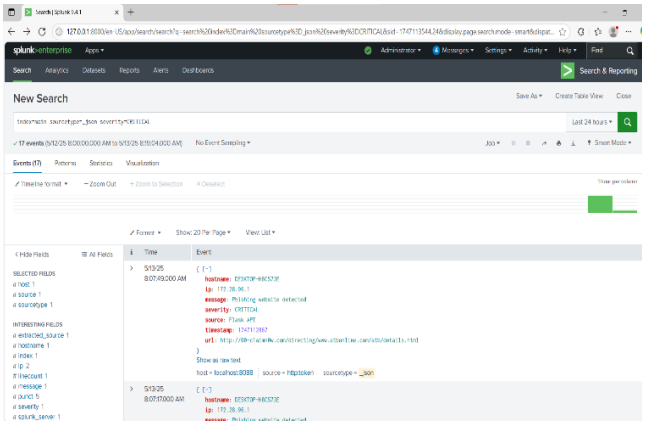

Fig. 10. Ingested Data

Fig. 11. Created Report

V. Conclusion And Future Works

To summarize, we proposed a machine learning model based on an SVM classifier trained on a dataset of 11,054 instances with 30 features. This model gained a 96.9% accuracy score. The model was then serialized and saved in a binary (.pkl) format via Python's pickle library and integrated with a backend server. Furthermore, the Chrome extension was connected to the Splunk tool via the backend server to ingest data about phishing website visit events.

For further enhancements, we recommend employing alternative machine learning classifiers or deep learning algorithms, as this could refine performance metrics. Additionally, user experience can be enhanced by incorporating feedback mechanisms and automatically capturing URLs. It is also recommended to develop a multi-browser extension to serve a broader range of users. Lastly, we suggest improving the reporting process by integrating email notifications within Splunk.

References

[1] M. Nadeem, S. W. Zahra, and M. N. Abbasi, 'Phishing Attack , Its Detections and Prevention Techniques', *Phishing Attack, Its Detect. Prev. Tech.*, vol. 1, no. 2, pp. 13–25, 2023, doi: 10.37591/IJWSN.

[2] 'Internet Crime Complaint Center, FBI'. [Online]. Available: https://www.fbi.gov/contact-us/field-offices/sanfrancisco/news/fbi-releases-internet-crime-report

[3] 'APWG Phishing Activity Trends Report, APWG.' [Online]. Available: https://apwg.org/trendsreports/

[4] M. N. Banu and S. M. Banu, 'A Comprehensive Study of Phishing Attacks', *A Compr. Study Phishing Attacks*, vol. 4, no. 6, pp. 783–786, 2013.

[5] S. Buchyk, D. Shutenko, and S. Toliupa, 'Phishing Attacks Detection', *CEUR Workshop Proc.*, vol. 3384, pp. 193–201, 2022.

[6] R. Alabdan, 'Phishing attacks survey: Types, vectors, and technical approaches', *Phishing Attacks Surv. Types, Vectors, Tech. Approaches*, vol. 12, no. 10, pp. 1–39, 2020, doi: 10.3390/fi12100168.

[7] A. A. Ubing, S. K. B. Jasmi, A. Abdullah, N. Z. Jhanjhi, and M. Supramaniam, 'Phishing website detection: An improved accuracy through feature selection and ensemble learning', *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 1, pp. 252–257, 2019, doi: 10.14569/IJACSA.2019.0100133.

[8] S. Kumar Gouda and M. Sandeepta Kumar, 'a Comprehensive Study of Phishing Attacks and Their Countermeasures', *A Compr. STUDY PHISHING ATTACKS THEIR Countermeas.*, no. May, p. 78, 2023, doi: 10.13140/RG.2.2.36686.13120.

[9] V. Bhavsar, A. Kadlak, and S. Sharma, 'Study on Phishing Attacks', *Study on Phishing Attacks*, vol. 182, no. 33, pp. 27–29, 2018, doi: 10.5120/ijca2018918286.

[10] N. Innab *et al.*, 'Phishing Attacks Detection Using EnsembleMachine Learning Algorithms', *Phishing Attacks Detect. Using Ensemble Mach. Learn. Algorithms*, vol. 80, no. 1, pp. 1325–1345, 2024, doi: 10.32604/cmc.2024.051778.

[11] A. Olayinka Taofeek, 'Development of a Novel Approach to Phishing Detection Using Machine Learning', *Dev. a Nov. Approach to Phishing Detect. Using Mach. Learn.*, vol. 12, no. 2, 2024, [Online]. Available: www.atbuftejoste.com.ng

[12] Vishesh Bharuka, Allan Almeida, and Sharvari Patil, 'Phishing Detection Using Machine Learning Algorithm', *Phishing Detect. Using Mach. Learn. Algorithm Vishesh*, vol. 10, no. 2, pp. 343–349, 2024, doi: 10.32628/cseit2410228.

[13] V. Vajrobol, B. B. Gupta, and A. Gaurav, 'Mutual information based logistic regression for phishing URL detection', *Mutual Inf. based Logist. Regres. phishing URL Detect.*, vol. 2, no. March, 2024, doi: 10.1016/j.csa.2024.100044.

[14] S. H. Ahammad *et al.*, 'Phishing URL detection using machine learning methods', *Phishing URL Detect. using Mach. Learn. methods*, vol. 173, no. November, p. 103288, 2022, doi: 10.1016/j.advengsoft.2022.103288.

[15] M. Sánchez-Paniagua, E. Fidalgo, E. Alegre, and R. Alaiz-Rodríguez, 'Phishing websites detection using a novel

multipurpose dataset and web technologies features', *Phishing websites Detect. using a Nov. Multipurp. dataset web Technol. Featur.*, vol. 207, no. June, p. 118010, 2022, doi: 10.1016/j.eswa.2022.118010.

[16] A. Aljofey *et al.*, 'An effective detection approach for phishing websites using URL and HTML features', *An Eff. Detect. approach phishing websites using URL HTML Featur.*, vol. 12, no. 1, pp. 1–19, 2022, doi: 10.1038/s41598-022-10841-5.

[17] S. Das Guptta, K. T. Shahriar, H. Alqahtani, D. Alsalman, and I. H. Sarker, 'Modeling Hybrid Feature-Based Phishing Websites Detection Using Machine Learning Techniques', *Model. Hybrid Featur. Phishing Websites Detect. Using Mach. Learn. Tech.*, vol. 11, no. 1, pp. 217–242, 2022, doi: 10.1007/s40745-022-00379-8.

[18] R. Yang, K. Zheng, B. Wu, C. Wu, and X. Wang, 'Phishing website detection based on deep convolutional neural network and random forest ensemble learning', *Phishing Website Detect. Based Deep Convolutional Neural Netw. Random For. Ensemble Learn.*, vol. 21, no. 24, pp. 1–18, 2021, doi: 10.3390/s21248281.

[19] A. Deshpande, O. Pedamkar, and N. Chaudhary, 'Detection of Phishing Websites Using Machine Learning', *Detect. Phishing Websites using Mach. Learn.*, vol. 10, no. 05, pp. 430–434, 2022, doi: 10.1109/ICCCI54379.2022.9740763.

[20] 'Phishing Website Detector, Kaggle.' [Online]. Available: https://www.kaggle.com/datasets/eswarchandt/phishing-website-detector

[21] A. Pradhan, 'SUPPORT VECTOR MACHINE-A Survey', *Support VECTOR Mach. Surv.*, vol. 2, no. 8, pp. 82–85, 2012.

[22] V. Apostolidis-Afentoulis, *SVM Classification with Linear and RBF kernels*. 2015. doi: 10.13140/RG.2.1.3351.4083.

[23] R. Abdillah, Z. Shukur, M. Mohd, and T. M. Z. Murah, 'Phishing Classification Techniques: A Systematic Literature Review', *Phishing Classif. Tech. A Syst. Lit. Rev.*, vol. 10, pp. 41574–41591, 2022, doi: 10.1109/ACCESS.2022.3166474.

[24] S. K. Satapathy, S. Mishra, P. K. Mallick, L. Badiginchala, R. R. Gudur, and S. C. Guttha, 'Classification of Features for detecting Phishing Web Sites based on Machine Learning Techniques', *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 8, pp. 425–430, 2019.

[25] N. S. Zaini *et al.*, 'Phishing detection system using machine learning classifiers', *Phishing Detect. Syst. using Mach. Learn. Classif.*, vol. 17, no. 3, pp. 1165–1171, 2020, doi: 10.11591/ijeecs.v17.i3.pp1165-1171.

[26] A. Khan, S. Hossen, A. Bhuiyan, and M. L. Hossain, 'Phishing Website Detection Using Machine Learning', *Phishing Website Detect. Syst. Using Mach. Learn.*, vol. 7, no. 2, 2024, doi: 10.1109/ICNWC60771.2024.10537279.

[27] L. Lakshmi, M. P. Reddy, C. Santhaiah, and U. J. Reddy, 'Smart Phishing Detection in Web Pages using Supervised Deep Learning Classification and Optimization Technique ADAM', *Wirel. Pers. Commun.*, vol. 118, no. 4, pp. 3549–3564, 2021, doi: 10.1007/s11277-021-08196-7.

[28] K. L. Chiew, C. L. Tan, K. Wong, K. S. C. Yong, and W. K. Tiong, 'A new hybrid ensemble feature selection framework for machine learning-based phishing detection system', *Inf. Sci. (Ny).*, vol. 484, pp. 153–166, 2019, doi: https://doi.org/10.1016/j.ins.2019.01.064.