Enhancing Phishing URL Detection Accuracy in Software-Defined Networks (SDNs) through Feature Selection and Machine Learning Techniques

A USHA RUBY
ausharuby@gmail.com

SRM Institute of Science and Technology

George Chellin Chandran J

Research Article

Keywords: Software-Defined Networks, URL, Phishing, Legitimate, Machine Learning

Posted Date: February 21st, 2024

DOI: https://doi.org/10.21203/rs.3.rs-3955168/v1

Additional Declarations: No competing interests reported.

Enhancing Phishing URL Detection Accuracy in Software-Defined Networks (SDNs) through Feature Selection and Machine Learning Techniques

Dr. A.Usha Ruby[1,*], Dr. George Chellin Chandran J[2]
Associate Professor[1, *], Director[2]
Computer Science and Engineering[1], SRMIST[1], Ramapuram[1], King's Academy[2]
ausharuby@gmail.com[1,*], chellin1968@gmail.com[2]

Abstract

Phishing attacks remain an enduring and ever-evolving menace to both networked systems and their users' privacy. In response to this formidable challenge, our research delves into an innovative approach designed to enhance the precision of phishing Uniform Resource Locator (URL) detection within the dynamic and programmable realm of Software-Defined Networks (SDNs). By harnessing feature selection capabilities and adaptive machine learning techniques, our proposed framework aims to fortify security measures in SDNs against these malicious campaigns. Our methodology's core is the deliberate selection of discriminative features from the extensive network data attributes. This feature selection process is meticulously designed to identify the most relevant characteristics associated with phishing URLs, thereby enabling the extraction of invaluable insights for more precise detection. These carefully chosen features then serve as inputs for a dynamic machine learning model, trained to adapt and evolve alongside the constantly changing landscape of phishing attacks. Within the SDN environment, our framework optimizes utilizing network resources and controller processing power. It achieves this by reducing the dimensionality of input data, resulting in improved detection accuracy and a decrease in false positives. The adaptive nature of our machine learning model ensures rapid recognition of emerging phishing tactics, thereby reducing the risk of succumbing to novel and sophisticated attacks. To validate the effectiveness of our approach, we conducted extensive experiments and evaluations within an SDN testbed, utilizing real-world phishing URL datasets. The results consistently demonstrate that our framework surpasses conventional methods, achieving higher detection accuracy and adaptability to evolving threats. In summary, our research represents a significant stride in the ongoing battle against phishing attacks by leveraging the dynamic capabilities of SDNs. The synergy between feature selection and adaptive machine learning techniques empowers SDNs to sustain accurate and effective phishing URL detection, ultimately reinforcing network security and safeguarding user privacy in an ever-evolving threat landscape.

Keywords: Software-Defined Networks, URL, Phishing, Legitimate, Machine Learning

I Introduction

In the digital era, where the internet is an integral part of daily life and business operations, the menace of phishing attacks remains a substantial concern. Phishing, a deceitful practice that tricks unsuspecting users into revealing sensitive information, has evolved into a highly sophisticated threat capable of inflicting significant financial losses and compromising data security. While traditional security measures have valiantly attempted to thwart phishing attempts, they often fall short in countering the constantly evolving tactics employed by cybercriminals. Consequently, the demand for innovative and adaptable solutions is more critical than ever. Software-defined networks (SDNs) have emerged as a transformative technology that empowers network administrators with unparalleled control and flexibility in managing network traffic. Their programmable and centralized nature renders SDNs an ideal platform for implementing advanced security measures. However, the effectiveness of phishing URL detection within SDNs can be further enhanced by integrating feature selection and adaptive machine-learning techniques. This journal paper embarks on a journey to explore how Feature Selection and Adaptive Machine Learning Techniques can be leveraged to strengthen the precision of phishing URL detection in SDN environments. By harnessing these methodologies, we aim to confront the dynamic nature of phishing attacks, which consistently evolve to circumvent conventional detection methods. In phishing URL detection, Machine Learning (ML) technology plays a crucial role alongside traditional methods like blacklisting and whitelisting. Many research studies have embraced ML as a viable approach [1,2], mainly because malicious URLs and deceptive web pages share common traits. However, a challenge arises when using all available URL features for ML training, as it can be time-consuming. Consequently, researchers have explored alternative techniques that focus on identifying specific phishing attributes in commonly encountered URLs while excluding irrelevant ones, leading to precise results. Advanced deep learning algorithms, including Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, showcase the ability to automatically identify meaningful patterns within datasets that conform to the classifier's criterion [3]. Additionally, various selection of feature techniques such as Particle Swarm Optimization [4], Feature Selection by Recursive

Elimination [5], Mine Blast [6], and SelectKBest [7] are employed to streamline the feature set and optimize phishing detection. This augmentation markedly improves the precision of phishing URL detection, eliminating the necessity to consider all available features. Moreover, the utilization of Feature (FSRE) in conjunction with K-means is applied to aid in data clustering. K-means, rooted in the principles of statistical learning theory, operates by minimizing structural risk, thereby reducing experimental errors and complexity. Its primary objective is to establish an optimal hyperplane with the maximum margin for classification [8]. Although these approaches improve the accuracy of identifying phishing URLs, they also bring about increased analytical costs and demand higher levels of hardware power consumption. As a result, these solutions might not be the most optimal selection for energy-efficient devices like wireless sensor networks [9,10].

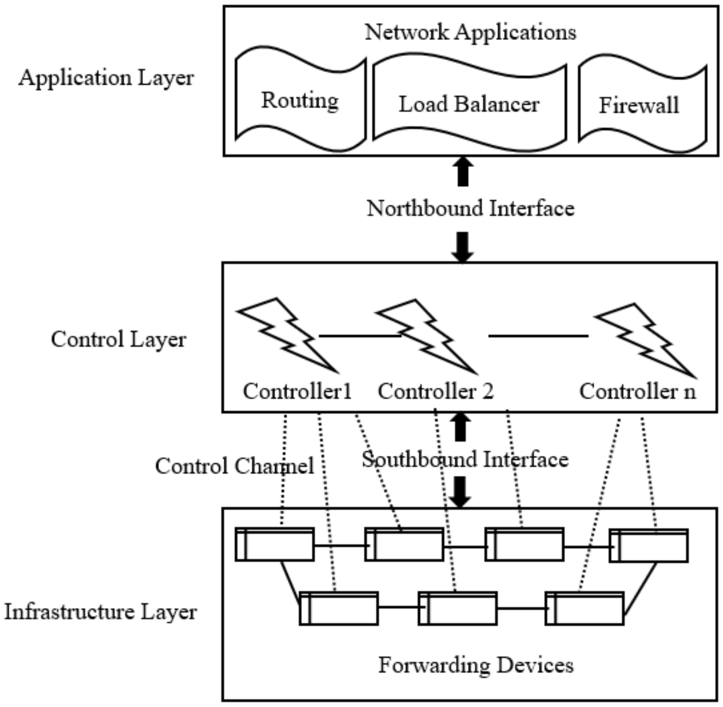


Figure 1. The conventional SDN framework

Software-defined networking (SDN) is poised to revolutionize future network management. This transformative approach involves decoupling control from forwarding devices, a well-established concept in prior research [11], as depicted in Figure 1. This transformation centralizes control units, providing them with comprehensive oversight over all network components and host devices. Consequently, these circumstances prove highly effective in countering phishing attacks across various host devices, irrespective of their processing capabilities. However, despite the potential of this approach in phishing detection, there remains a scarcity of studies that have harnessed its capabilities in the existing literature. While previous authors [12,13] have integrated SDN technology with machine learning algorithms to enhance phishing site detection, these efforts were typically in conjunction with whitelisting and blacklisting technologies.

In this research endeavor, our aim is to advance the field of phishing URL detection through the integration of SDN (Software-Defined Networking), feature selection, and clustering techniques (FSRE, K-means) with Convolutional Neural Network (CNN) algorithms. This amalgamation is designed to augment accuracy while concurrently reducing processing time. In this configuration, the SD-Controller, seamlessly integrated into the SDN architecture, assumes responsibility for handling the Feature Selection CNN (FS-CNN). The results generated by FS-CNN are subsequently relayed to the SD-Switches via the adjustment of flow table conventions. Whenever new packets are dispatched from hosts to SD-Switches, they undergo preprocessing and evaluation against the flow table rules, determining whether they should be permitted to advance or discarded. The FS-CNN encompasses several sequential steps, including preprocessing, feature matrix extraction, feature learning, and classification. A significant aspect of our research lies in its independence from the necessity of retrieving content from target websites or relying on third-party services. Instead, it focuses on capturing information and sequential patterns within URL strings, thus obviating the requirement for prior knowledge of phishing and enabling swift classification of actual URLs. Our study is committed to enhancing network security in the face of the persistent threat of phishing by presenting a comprehensive approach that merges intelligent feature selection with adaptive machine learning algorithms.

The key impacts of this research can be concisely outlined as follows:

Introduction of an innovative URL phishing recognition framework, FS-SDN-CNN, which effectively harnesses embedded SDN technology in tandem with the deep learning Convolutional Neural Network (CNN) algorithm. This pioneering approach optimizes resource utilization by shifting the training burden for detection to the SD controller. The creation of an innovative deep learning method, FS-SDN-CNN, seamlessly amalgamates feature extraction using binary encoding, Feature Selection through Recursive Elimination clustering (FSRE-K-means), and the CNN algorithm to significantly boost detection accuracy. A comprehensive evaluation of FS-SDN-CNN employing real-world datasets, accompanied by a comparative analysis against existing methodologies. Furthermore, the real-world efficiency of FS-SDN-CNN is rigorously assessed.

The subsequent sections of this manuscript are structured as follows:

Section 2 furnishes an outline of associated research in the realms of SDN, machine learning techniques, and feature selection.

Section 3 elucidates the methodology of FSRE-K-means-CNN-SDN, providing a comprehensive understanding of its structure and operational mechanisms.

In Section 4, we delve into discussions regarding the empirical setup, performance measures, and the presentation of our findings.

Lastly, Section 5 encapsulates the conclusion of our study and highlights potential avenues for upcoming research.

II Literature Review

The detection and prevention of phishing attacks have been perennial challenges in the realm of cybersecurity, with phishing websites posing a significant threat to users' sensitive information and digital security. In this context, a substantial body of research has emerged, aiming to develop effective methods for identifying and mitigating phishing websites. Phishing attacks typically involve malicious actors creating deceptive web pages that mimic legitimate websites, aiming to trick users into divulging their personal and confidential information. Traditional phishing detection methods have often relied on blacklists, heuristics, or URL-based techniques. While these approaches have had some success, they are limited in their ability to detect novel and evolving phishing sites. The paper by Zhang et al. (2007) [14] introduces a novel content-based approach to phishing detection, known as Cantina. This approach represents a paradigm shift in combating phishing threats by focusing on the content and visual elements of web pages rather than relying solely on URL or domain reputation. Cantina leverages machine learning and feature extraction techniques to analyze the content of web pages, identifying patterns and characteristics commonly associated with phishing attempts. S. Ustebay et al. (2019) [15] employ Feature Selection by Recursive Elimination to enhance efficiency by selecting the most relevant features while eliminating less informative ones. They integrate the robust Random Forest classifier for accurate intrusion detection and leverage Deep Learning to capture complex patterns and adapt to evolving threats. Empirical validation demonstrates the effectiveness of this combined approach, highlighting its potential to bolster intrusion detection systems. R. Wazirali, and R. Ahmad (2022) [16] center on the detection of Denial-of-Service (DoS) attacks, a pervasive threat to WSNs. By leveraging machine learning, the study strives to improve the robustness of DoS detection mechanisms in WSNs, ultimately contributing to the sustainability and reliability of these critical network systems.

The significance of this work by S. Venkatraman (2019) [17] lies in its potential to provide more robust protection against evolving threats by leveraging the capabilities of deep learning. It underscores the importance of harnessing the power of deep learning to tackle the increasingly sophisticated and adaptive nature of modern cyber threats, making it an important resource for the field. W. Wei et al. 2020 [18] underscore the importance of its potential to bolster online security by harnessing deep learning capabilities. By introducing a CNN-based model, the research addresses the growing concern of phishing attacks, which are a prevalent threat to online users. The paper's emphasis on both accuracy and speed highlights its practicality in real-time applications.

The study by M. Miao and B. Wu's (2020) [19] significance lies in its adaptability and flexibility in countering phishing threats. By integrating SDN and ensemble learning, the research offers a dynamic and effective solution for identifying phishing attacks, which are notorious for their ability to rapidly evolve and bypass traditional methods. This work underscores the importance of embracing emerging technologies like SDN and harnessing the power of ensemble learning to create versatile and responsive tools in the ongoing battle to protect against phishing attempts.

In 2019, Abdullaziz OI and Wang L conducted research focused on enhancing the security of Software-Defined Networking (SDN) controllers by thwarting Denial of Service (DoS) attacks through the implementation of information-hiding techniques. Protecting SDN controllers is of paramount importance, given their central role in SDN management. This study likely delves into strategies for concealing sensitive information and fortifying SDN controllers against various forms of DoS attacks [20].

In a separate study published in 2017, Agborubere B and Sanchez-Velazquez tackled the realm of OpenFlow communications and Transport Layer Security (TLS) within SDNs. OpenFlow serves as a cornerstone protocol within SDNs, and safeguarding its communication channels is of utmost significance. This research likely explores the utilization of TLS to encrypt OpenFlow communications, thus bolstering overall security [21].

Ahmad I et al., (2015) provided a survey of security issues in SDNs. It is likely to cover various aspects of security within SDNs, including vulnerabilities, threats, and existing solutions. It should serve as a comprehensive overview of the field [22].

Ahmed ME, and Kim H (2017) focused on the Internet of Things (IoT), this paper addresses Distributed Denial of Service (DDoS) attack mitigation using SDN. Given the proliferation of IoT devices, securing them with SDN is essential. The paper might discuss how SDN can be leveraged to defend against DDoS attacks in an IoT context [23].

Aizuddin AA et al., (2017) discussed DNS (Domain Name System) amplification attack detection and mitigation using SDN. DNS amplification attacks are a common threat, and SDN can be employed to detect and prevent such attacks effectively. The paper might detail the implementation of sFlow-based SDN solutions for this purpose [24].

In 2017, Al-Haj S and Tolone WJ addressed the issue of misconfigurations within the flow table pipelines of Software-Defined Networks (SDNs). These misconfigurations have the potential to expose vulnerabilities and introduce inefficiencies into network operations. This paper is likely to delve into approaches for identifying and mitigating such misconfigurations within SDN environments [25].

In 2018, Alasadi E and Al-Raweshidy HS introduced Servers under Software-Defined Network Architectures as a solution aimed at reducing the presence of discovery messages. By minimizing unnecessary network traffic, this innovation seeks to enhance both network performance and security. The paper likely provides insights into the implementation and advantages of the SSED approach [26].

In 2014, Alcorn JA and Chow CE presented a framework designed for modeling and simulating attacks on OpenFlow networks. Such simulations play a pivotal role in evaluating network security and readiness. This paper probably details the framework's design and its practical applications in assessing network security [27].

In 2018, Allouzi M. and Khan J. focused their attention on Safe Flow, an authentication protocol tailored for SDNs. Authentication holds significant importance in ensuring network security, and this paper may shed light on how SafeFlow reinforces security measures within software-defined networks [28].

In 2017, Alparslan O et al. addressed the issue of resilience against DDoS attacks using multipath orchestration and SDN of Virtual Network Function services. This paper likely outlines strategies for mitigating DDoS attacks through the utilization of SDN-based methodologies [29].

In the same year, Ambrosin M et al. introduced Line Switch as a solution designed to combat control plane saturation attacks within SDNs. These control plane attacks have the potential to disrupt network management functions. The paper may elaborate on how Line Switch tackles this challenge [30].

Also in 2017, Aseeri A et al. focused on countering eavesdropping attacks within the SDN data plane. Securing the data plane is essential for safeguarding network traffic, and this paper is likely to discuss methods for preventing eavesdropping incidents in SDN environments [31].

In 2017, De Assis MVO et al. introduced a game-theoretical-based system employing various algorithms to mitigate DoS and DDoS attacks in SDN networks. This paper is expected to explore the synergistic application of these techniques for effective attack mitigation [32].

The paper by Yang et al. (2019) [33] presents a novel approach to phishing website detection using deep learning and multidimensional features. Their use of convolutional and recurrent neural networks demonstrates high accuracy in distinguishing phishing websites from legitimate ones. However, future research should focus on addressing computational resource limitations and adapting the model to evolving phishing tactics.

Rao et al. (2020) [34] tackle phishing website detection by focusing on URL inspection. Their novel approach emphasizes the analysis of URL features to identify potential phishing threats. Although the paper provides promising results in detecting phishing websites based on URL characteristics, further research is needed to assess its performance across a broader range of phishing tactics and the integration of complementary detection methods to enhance its overall effectiveness.

Chiew et al. (2019) [35], the authors introduce a novel approach to enhance phishing website detection. Their hybrid ensemble feature selection framework combines the strengths of multiple classifiers and feature selection methods, addressing a critical aspect of machine learning-based cybersecurity.

Zhu et al. (2019) [36], the authors propose a phishing detection model that combines optimal feature selection and neural networks. This paper enhances the domain of cybersecurity by emphasizing the importance of feature selection in enhancing detection accuracy. The paper's results suggest that the OFS-NN model offers an effective approach to phishing website detection, showcasing its potential for improving online security.

Mao et al. (2018) [37] introduce a novel approach to phishing detection by analyzing page layouts. This research explores a unique perspective in the field of cybersecurity, emphasizing the importance of visual and layout-based features in identifying phishing websites. The paper's findings suggest that aggregating page layout information can be an effective method for enhancing phishing detection, offering potential improvements to online security.

Sahingoz et al. (2019) [38] present an ML approach for detecting phishing URLs. This paper enhances the domain of cybersecurity by emphasizing the significance of URL analysis. The study's findings indicate that machine learning techniques applied to URL features can be effective in identifying phishing attempts, potentially enhancing online security measures.

Bahnsen et al. (2017) [39] employ recurrent neural networks (RNNs) to classify phishing URLs. This study is significant for cybersecurity as it demonstrates the applicability of advanced neural network models to tackle the challenge of identifying phishing websites. The paper showcases that RNNs can be effective tools for enhancing the detection and prevention of online phishing threats, offering valuable insights for cybercrime research and mitigation efforts. In conclusion, the reviewed literature underscores the critical importance of enhancing phishing URL detection accuracy in the context of SDNs. The incorporation of feature selection methods and adaptive machine learning techniques represents a significant step forward in strengthening cybersecurity methods. These approaches not only demonstrate the potential to increase detection accuracy but also offer adaptability to evolving phishing tactics, a crucial attribute in the ever-changing threat landscape. As organizations continue to rely on SDNs for their network infrastructures, the integration of these advanced methodologies promises to provide robust protection against phishing threats, safeguard sensitive information, and maintain network integrity.

III Proposed FSRE-K-means-CNN-SDN method

In this segment, we introduce the FSRE-K-means-CNN-SDN architecture, a novel approach tailored for the precise detection of phishing websites. It harnesses a combination of feature selection methodologies, deep learning techniques, SDN, and to substantially improve the precision of phishing URL detection. The main objective behind the integration is to establish an ongoing training process capable of effectively countering modern phishing attacks. This approach not only enhances URL detection accuracy but also shifts the responsibility for implementation and training costs from end-user devices to the networking infrastructure, specifically the SDN controller and switches. Within this architecture, the Convolutional Neural Network (CNN) [40] operates within the SD-Controller, facilitating the crucial training process. The outcomes of this process are then transmitted to the SD-Switches through the OpenFlow protocol, where they are applied to incoming packets originating from user devices. Figure 2 provides a comprehensive logical diagram illustrating our proposed model. It's essential to highlight that our model is uniquely engineered for operation within the network ecosystem of an Internet Service Provider (ISP) employing SDN technology. In this arrangement, the SD-Controller plays a critical role in filtering URL packets across all user devices. As depicted in Figure 2, the FSRE-K-means-CNN method functions within the realm of the SD-Controller. In this setup, every user-initiated URL request passes through the SD-Switches before reaching the internet, and conversely, the SD-Controller serves as the intermediary. The primary responsibility of the controller is to meticulously evaluate each user's URL request, determining its legitimacy or the possibility of it being a phishing website. This task is accomplished by harnessing the capabilities of the SD-Controller through the deployment of the FSRE-K-means-CNN technique. This strategic approach effectively offloads the responsibility of detecting URL anomalies from individual user devices to the centralized SD-Controller, offering advantages for devices with constrained computational memory and power resources.

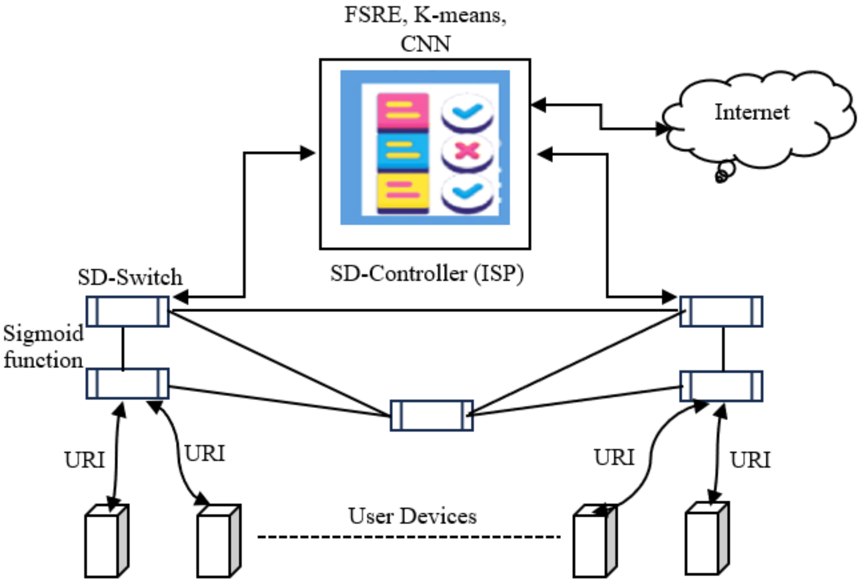


Figure 2. Proposed Model

The FSRE-K-means-CNN framework comprises a series of sequential components, which include preprocessing, feature learning, feature matrix extraction, and classification. Each URL is assessed to verify its legitimacy or the likelihood that it is a phishing website after these steps are completed. The controller then applies the most recent sigmoid function parameters to the flow entries in the switches in preparation for later flow matching. The procedure starts when a user's data packet enters the switch. The switch starts a check to see if the URL packet flow has already been registered. The sigmoid function is then used following this confirmation. The data packet is transmitted immediately if the result is favourable. The switch, however, sends the URL packet to the controller for additional processing using the FSRE-K-means-CNN method if the outcome is unfavourable. Subsequent sections will provide a more comprehensive exploration of the intricacies of the FSRE-K-means-CNN scheme.

The FSRE-K-means Model

In this model, our primary objective is the categorization of URLs as either legitimate or phishing, with a paramount focus on achieving an exceptionally high detection accuracy for phishing sites. Despite the potential high cost associated with phishing detection, the utilization of SDN technology underscores the importance of accuracy. Figure 3 presents a representation of the components of the FSRE-K-means-CNN model.

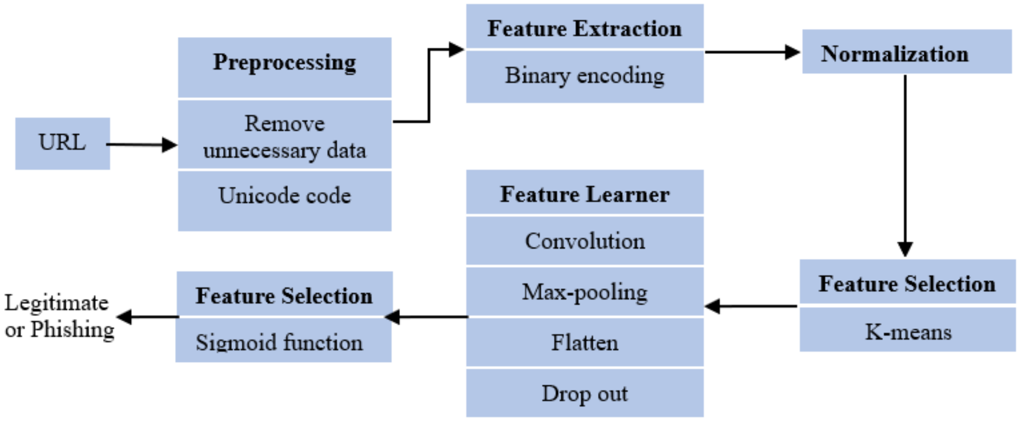


Figure 3. FSRE-K-means-CNN

The FSRE-K-means-CNN framework adheres to a sequential process that encompasses several crucial phases: URL preprocessing, feature matrix extraction, feature selection, normalization, feature learning, and classification.

URL Composition

A URL comprises six interrelated components that precisely specify the resource's location, as explained in Figure 4. These components include the subdomain, second-level domain, protocol, path information, filename, and top-level domain.

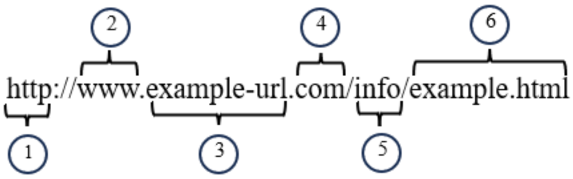


Figure 4. URL components

Phishers frequently endeavour to fabricate phishing URLs that closely mimic legitimate sites, aiming to deceive online users. However, legitimate website URLs are generally well-documented, and attackers cannot simply replicate them. Instead, they create URLs that bear a resemblance to the original ones. Consequently, distinguishing between legitimate and phishing URLs relies on an array of URL features. In the dataset under consideration, we leverage a set of 30 distinct features for this differentiation.

Preprocessing

The dataset under consideration, obtained from online sources, contains a significant quantity of URLs categorized as either legitimate or phishing. In the process of creating an effective feature matrix, we employ a preprocessing step to remove superfluous and unclear characters. Specifically, ambiguous characters like the subdomain (www) and protocol (https or http) are methodically eliminated from the URL strings prior to extracting the feature matrix. Additionally, we standardize the URL by converting all uppercase characters to lowercase, ensuring uniform treatment of alphabetic characters throughout our method.

Binary Encoding – Feature Extraction

Dimensionality reduction methods utilized in this research entail counting the characters found in URLs. URLs contain a range of characters, including letters (A-Z), digits (0-9), and certain alphanumeric symbols like '@', '#', '%', '$', and so on. Each character's frequency in the URLs that have been processed is quantified to create the feature matrix. To illustrate, when extracting features from "https://www.google.com," the feature matrix is generated without incorporating any ambiguous characters. Additionally, any characters outside the previously mentioned set are disregarded. Binary Encoding [41] is a technique that can be particularly useful when dealing with high-cardinality categorical features like URLs and domain names in the context of phishing detection. Here's why Binary Encoding can be a suitable choice: Dimensionality Reduction: Binary Encoding reduces the dimensionality of the categorical features compared to One-Hot Encoding. Instead of creating a binary feature for each unique category, it encodes each category as a binary code, typically using log2(N) bits, where N is the number of unique categories. This can significantly reduce the number of features, which can be especially beneficial when dealing with many categories. Efficient Representation: Binary Encoding efficiently represents categorical data in a compact binary format. Each category is mapped to a binary string, where each bit represents the presence or absence of a specific category. This compact representation can lead to faster model training and reduced memory usage. Retains Information: Binary Encoding retains information about the relationships between categories to some extent. Since categories are encoded as binary codes, they capture certain similarities or differences between categories based on their binary representations. Interpretability: While not as directly interpretable as One-Hot Encoding, Binary Encoding still allows you to analyze the importance of different categories in the phishing detection process by examining the binary patterns associated with each category.

Data Normalization

Normalization serves to enhance the feature matrix's quality by rescaling it within the range of 0 to 1. For this purpose, we employ the min-max normalization method [42], as specified by the following equation (1):
scaled_value = (original_value - min) / (max - min) (1)
This normalization procedure ensures that the features are scaled appropriately for subsequent analysis and modeling.

Feature Selection Technique and Clustering

To reduce the dimensionality of the feature matrix, this step involves identifying related features. Given that the dataset used in this study pertains to linear relationships and statistical learning, the Feature Selection by Recursive Elimination (FSRE) [43] method with k-means clustering is proposed for this purpose. Mutual information measures the dependency between two variables, and in the context of feature selection, it quantifies the amount of information shared between a feature and the target variable. In phishing detection, mutual information can be used to identify the most relevant features (e.g., attributes of URLs or website content) that are informative for distinguishing between phishing and non-phishing instances.

Mutual information be applied for feature selection in a phishing dataset is as follows:
1. Calculate the mutual information between each feature (attribute) and the target variable (phishing vs. non-phishing). High mutual information indicates strong dependence and relevance.
2. Rank the features based on their mutual information scores.
3. Select the top-ranked features as the most informative ones for building your phishing detection model.

Clustering Method: K-means Clustering

K-means Clustering: K-means is a popular unsupervised clustering algorithm used to group similar data points into clusters. In the context of phishing detection, K-means [44] can be applied to identify patterns or clusters among URLs or network traffic data. For instance, K-means can help identify groups of URLs that exhibit similar characteristics, which may include phishing patterns.

Steps of K-means clustering to a phishing dataset:

1. Prepare the dataset by extracting relevant features from the URLs or network traffic data.
2. Normalize or scale the features.
3. Choose the number of clusters (k) based on domain knowledge or by using techniques like the elbow method or silhouette analysis.
4. Apply K-means clustering to the feature vectors (representing URLs or network data).
5. Examine the clusters to identify any patterns that may indicate groups of URLs with similar characteristics, which could be indicative of phishing behaviour.

The feature subset with the greatest weight scores is next submitted to K-means clustering, which classifies features using a linear kernel.

Feature Learner

The process of extracting features within the FSRE-K-means-CNN framework involves the utilization of a Convolutional Neural Network (CNN) with specific enhancements tailored for this proposal. This feature learning component encompasses several essential elements: a flatten layer, a convolutional layer, a max-pooling layer, and a dropout layer. The structural configuration of the feature learner is visually depicted in Figure 5.

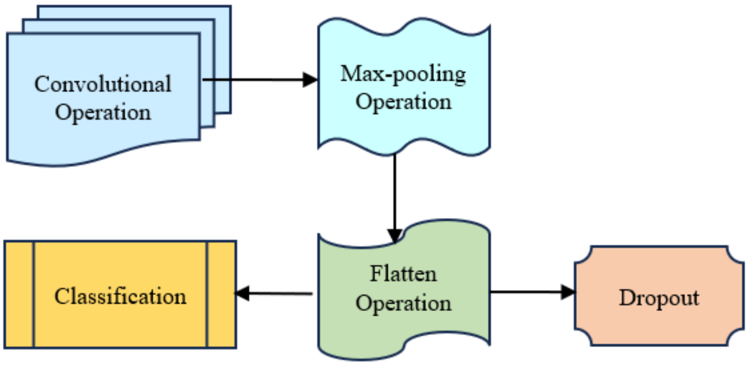


Figure 5. Feature Learner

The primary function of the convolutional layer is to carry out the convolution operation, which denotes the combining of the feature matrix with a predetermined starting filter [45]. The Rectifier Linear Unit (ReLU) activation function is employed to merge the input set with the feature matrices, as documented in reference [46]. It's worth noting that the choice of activation function, such as ReLU, is pivotal in representing the feature matrix within hierarchical Neural Networks (NNs). ReLU is particularly renowned for its ability to promote sparsity in the matrix. The mathematical description for the convolutional function is provided in equation (2) and the ReLU activation function is provided in equation (3).

$$s[y] = (z * p)[y] \quad (2)$$

$$s[y] = \max(z, s[y]) \quad (3)$$

Here, s[y] represents the feature map, z denotes attribute data, and p signifies kernels. Following the processing of the input by the convolutional layer, the next stage involves the max-pooling layer, which plays a pivotal role in selecting the most relevant outputs. Max-pooling introduces local invariance, effectively downsizing the input elements [46]. This step significantly enhances the performance of the convolutional process by identifying the essential features. Within the subsequent convolutional layer, a reduction in the complexity of the feature maps is implemented. This reduction is accomplished through the application of a max-pooling layer with a pool size of 2, which was selected based on experimentation involving different pool sizes, ultimately yielding the most favorable results. Moving on to the next layer, a flattening operation is employed to convert the feature maps into a vector form. During the classification phase, this flattened input vector is passed through the network to generate numerical outputs in each output neuron [47]. Essentially, this process transforms the feature representation into a single extended layer that originates from the convolutional layer. To prevent overfitting in fully connected layers, a dropout layer is incorporated. This dropout layer helps maintain the model's generalization capabilities by randomly dropping a fraction of neurons during training, preventing excessive reliance on specific neurons, and enhancing the network's robustness.

Classification

The feature map undergoes classification as either phishing or legitimate in the ultimate classification layer through the utilization of the sigmoid activation function with equation 4.

$$s[p] = 1 / (1 + e^\wedge-(w^{t} * x + b)) \qquad (4)$$

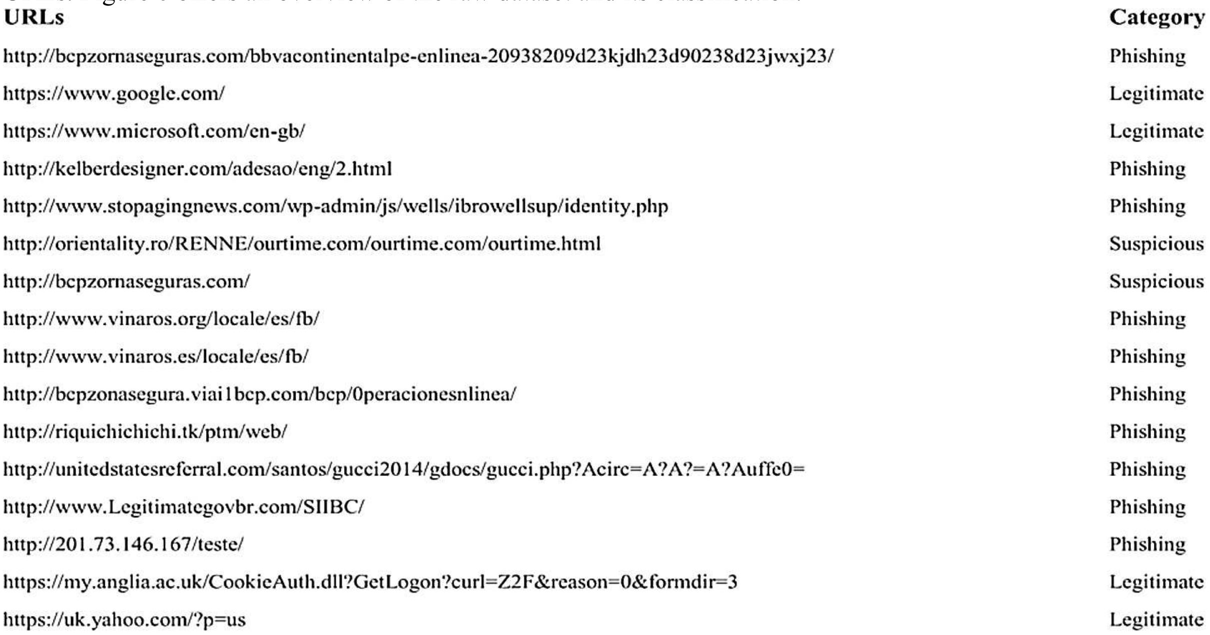
The determination of the classification is contingent on the value of s[p], which ranges from 0 to 1. When s[p] is less than or equal to 0.5, the URL is categorized as phishing, while values greater than 0.5 result in its classification as legitimate. Crucially, the controller continuously updates the parameters of the sigmoid function within the switch via the Packet_In message. Consequently, the switch proceeds to update its flow table. As a result, when new URL packets reach the switch, they undergo preprocessing and receive a label based on the sigmoid function. The specifics of the URL inspection process are delineated in the Algorithm.

```
Algorithm: Examination of URLs in SD-Switch
Input: URL packet.
Perform preprocessing on the URL packet.
Apply binary transformation to the URL packet.
Normalize the URL packet.
Obtain the attribute from the URL packet.
Output: Sigmoid activation function implementation.
If the output is less than or equal to 0.5, then
Discard the packet.
Otherwise,
Route the packet to the Internet.
End if
```

This algorithm outlines the process of inspecting and classifying URL packets within the SD-Switch. It begins by receiving a URL packet and then proceeds through a series of preprocessing steps, including binary encoding and normalization. Subsequently, it extracts the attribute data (x) from the URL packet and applies a sigmoid activation function. Based on the output of the sigmoid activation function, the algorithm decides. If the output is less than or equal to 0.5, indicating a high likelihood of phishing, the packet is dropped to prevent access to a potentially malicious website. If the output is greater than 0.5, indicating a lower likelihood of phishing, the packet is forwarded to the Internet, allowing access to the requested website. This algorithm facilitates the real-time inspection and classification of incoming URL packets, helping to protect users from potentially harmful websites.

IV Experimental Results

In this section, we will evaluate the model's performance through simulation software and make comparisons with recent research findings. The datasets utilized for this evaluation were compiled from diverse sources. Phishing URLs were obtained from https://www.phishtank.com during the same period, while genuine URLs were extracted from https://5000best.com/websites [48]. The dataset encompasses a total of 51,100 URL samples, comprising approximately 39,000 legitimate URLs and the remaining samples corresponding to phishing URLs. Figure 6 offers an overview of the raw dataset and its classification.



URLs	Category
http://bcpzornaseguras.com/bbvacontinentalpe-enlinea-20938209d23kjdh23d90238d23jwxj23/	Phishing
https://www.google.com/	Legitimate
https://www.microsoft.com/en-gb/	Legitimate
http://kelberdesigner.com/adesao/eng/2.html	Phishing
http://www.stopagingnews.com/wp-admin/js/wells/ibrowellsup/identity.php	Phishing
http://orientality.ro/RENNE/ourtime.com/ourtime.com/ourtime.html	Suspicious
http://bcpzornaseguras.com/	Suspicious
http://www.vinaros.org/locale/es/fb/	Phishing
http://www.vinaros.es/locale/es/fb/	Phishing
http://bcpzonasegura.viai1bcp.com/bcp/0peracionesnlinea/	Phishing
http://riquichichichi.tk/ptm/web/	Phishing
http://unitedstatesreferral.com/santos/gucci2014/gdocs/gucci.php?Acirc=A?A?=A?Auffe0=	Phishing
http://www.Legitimategovbr.com/SIIBC/	Phishing
http://201.73.146.167/teste/	Phishing
https://my.anglia.ac.uk/CookieAuth.dll?GetLogon?curl=Z2F&reason=0&formdir=3	Legitimate
https://uk.yahoo.com/?p=us	Legitimate

Figure 6. Unrefined web links alongside their respective categorization tags.

Additionally, we conduct a comparative analysis between our approach and prior studies [1, 9, 10, 36, 40] utilizing the FSRE-K-means-CNN technique.

Simulation Settings

Our simulations were conducted using the Open Network Operating System (ONOS) controller [49] and the Mininet tool [50]. We configured FSRE-K-means-CNN parameters as detailed in Table 1. The second dense layer has an output shape of (None, 2). The first dense layer outputs data in the shape of (None, 16). Another dense layer outputs data with the shape of (None, 32). The momentum value for optimization is set to 0.5. Max pooling is applied with a shape of (None, 26, 64). The learning coefficient ranges from 0.1 to 0.01, allowing for adaptive adjustments during training. The minibatch size is specified as 45. A 1D convolutional layer is used with an output shape of (None, 52, 64). The kernel filter size is set to 3 x 3. Dropout is applied with an output shape of (None, 32). Max pooling is used once again with a shape of (None, 26, 64). The flatten layer transforms the data to the shape of (None, 1664). The batch size for training is set to 32. The model is trained for a total of 200 epochs with these parameter settings.

Table 1. Configuration of parameters for the FSRE-K-means-CNN

Parameter	Value
Dense 2	(None, 2)
Dense 1	(None, 16)
Dense	(None, 32)
Momentum	0.5
Max pooling	(None, 26, 64)
Learning coefficient	Ranging from 0.1 to 0.01
Minibatch	45
Conv1D	(None, 52, 64)
Kernel Filter	3 x 3
Dropout	(None, 32)
Max pooling	(None, 26, 64)
Flatten	(None, 1664)
Batch size	32
Epochs	200

Performance Metrics

In this research, we utilize five performance metrics to assess FS-SDN technology. Our primary objectives revolve around maximizing the count of True Negatives (TN) and True Positives (TP) while minimizing False Negatives (FN) and False Positives (FP). TN signifies the number of legitimate URLs accurately recognized, while TP indicates the count of phishing URLs correctly classified. Moreover, FN signifies the count of regular URLs incorrectly categorized as phishing, while FP represents the count of legitimate URLs mistakenly identified as phishing URLs. We use the following performance metrics:

Recall is defined as the proportion of correctly retrieved relevant items (TP) relative to the total number of relevant items (TP + FN). Recall serves as a metric to gauge the model's effectiveness in identifying relevant items accurately.

Recall = TP / (FN + TP)   (5)

Precision measures the accuracy of phishing URL detection. It is the ratio of correctly identified phishing URLs (TP) to the total number of identified phishing URLs (TP + FP).

Precision = TP / (TP + FP)   (6)

Accuracy calculates the percentage of correct predictions made by the model among all predictions, encompassing both legitimate and phishing URLs.

Accuracy = (TN + TP) / (FP + FN + TP + TN)   (7)

The F1 Score offers a comprehensive measure of model accuracy by taking both precision and recall into account. It is the harmonic mean of precision and recall and is computed as:

F1 Score = 2 * (Recall * Precision) / (Recall + Precision)   (8)

In addition to these metrics, our evaluation of FS-SDN includes the consideration of packet processing time. We use the Percentage Value (PV) to assess accuracy and the F1 score about the optimal value of 1. Additionally, we evaluate the False Positive Rate (FPR) and False Negative Rate (FNR) relative to the optimal value of 0. Furthermore, our performance metrics encompass memory usage and the number of URLs processed per second. Collectively, these metrics provide a comprehensive assessment of the FS-SDN model's performance, encompassing accuracy, efficiency, and resource utilization.

The dataset used in our research consists of around 51,000 URLs, spanning a range of lengths from 15 characters to over 200 characters. To evaluate how URL length affects the performance metrics of FSRE-K-means-CNN, we conducted a focused analysis specifically considering URLs within the length range of 15 to 200 characters. Notably, most of these URLs fall within the range of 10–60 characters in length, followed by an average between 61 and 100 characters, and finally, some URLs extend from 101 to 200 characters. Figure 7 provides a graphical representation of these analyses. As illustrated in Figure 7, the performance metrics of FSRE-K-means-CNN, including accuracy, precision, recall, and F1-score, reveal unique distributions across different URL lengths. Each specific URL length (denoted as 'L') exerts a distinctive influence on these four metrics. In addition, Table 2 offers a more comprehensive presentation of the findings, depicting the variations in performance metrics across varying URL lengths. It is evident from this discussion that the length of a URL (denoted as 'L') should ideally fall within a balanced range, avoiding extremes, as revealed by the empirical results. Accuracy measures the overall correctness of a model's predictions. As we examine the table, we can observe a general trend where accuracy tends to increase as the input length grows from 10 to 60, with the highest accuracy of 0.9965 achieved at a length of 55. However, after that point, there's a slight decline, but the model remains highly accurate. At length 200, accuracy is slightly lower at 0.9831. Precision reflects the model's ability to make accurate positive predictions. In this table, precision mostly maintains a consistent performance across different lengths. It consistently remains above 0.985, indicating that the model effectively avoids false positives in its predictions. Recall signifies the model's capability to correctly identify positive instances. Like precision, recall remains relatively stable across various lengths. It exhibits a peak value of 0.9936 at a length of 40 but doesn't deviate significantly from this value at other lengths. The F1 Score is a balance between precision and recall, providing an overall assessment of a model's performance. It follows a pattern-like accuracy, with an initial increase as length grows, a peak value of 0.9937 at length 40, and a subsequent slight decrease. Even at its lowest point

(0.9821 at length 200), the F1 Score remains relatively high. In summary, this comparison highlights that the model performs exceptionally well across a range of input lengths, particularly in terms of precision and recall. While accuracy and the F1 Score exhibit some fluctuations with varying input lengths, the model's performance remains impressive, demonstrating its versatility and effectiveness in handling diverse data lengths.

Table 2. Performance Analysis of FSRE-K-means-CNN across various URL Lengths

Length	Accuracy	F1 Score	Recall	Precision
10	0.9849	0.9869	0.9849	0.9855
15	0.9855	0.9867	0.9864	0.9865
20	0.9877	0.9878	0.9877	0.9882
25	0.9881	0.9884	0.9883	0.9872
30	0.9942	0.9906	0.9923	0.9934
35	0.9905	0.9922	0.9931	0.9952
40	0.9931	0.9937	0.9936	0.9901
45	0.9952	0.9936	0.9935	0.9913
50	0.9944	0.9913	0.9911	0.9915
55	0.9965	0.9911	0.9916	0.9911
60	0.9918	0.9929	0.9938	0.9949
75	0.9923	0.9921	0.9922	0.9941
85	0.9942	0.9923	0.9924	0.9922
95	0.9915	0.9923	0.9925	0.9912
115	0.9924	0.9924	0.9923	0.9921
135	0.9851	0.9886	0.9877	0.9882
165	0.9863	0.9835	0.9832	0.9825
200	0.9831	0.9821	0.9831	0.9833

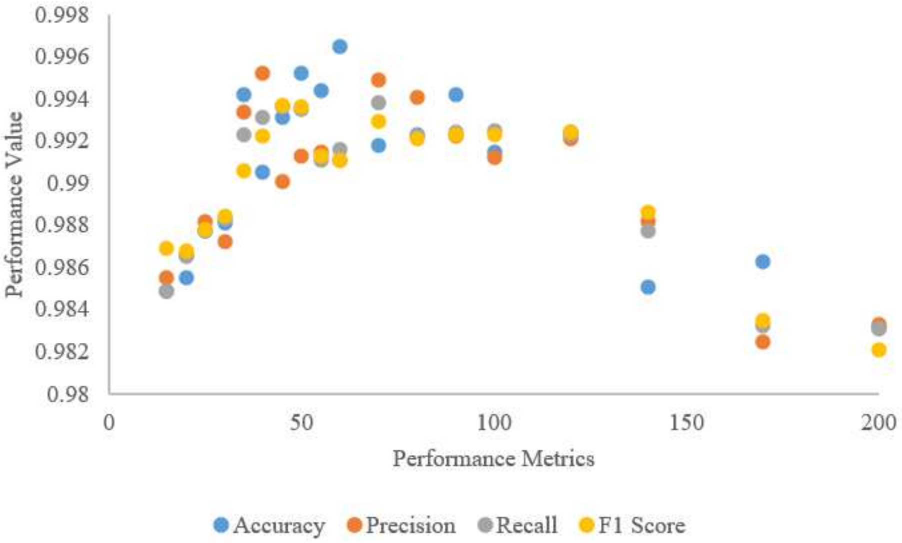


Figure 7. Distribution of FSRE-K-means-CNN Performance Metrics in Relation to Varied URL Lengths

Moreover, the selection of the feature selection method is another pivotal factor that significantly impacts the performance of FSRE-K-means-CNN. To elucidate the rationale behind adopting FSRE-K-means, we conducted an evaluation of various feature selection techniques with respect to FSRE-K-means-CNN's performance metrics. These techniques encompass SelectKBest-Chisquare [51], SelectKBest-Anova, SelectFromModel-L1 [52], and FSRE-K-means. Figure 8 visually portrays the performance metrics of these algorithms, employing the same dataset and FSRE-K-means-CNN parameter settings for consistency.

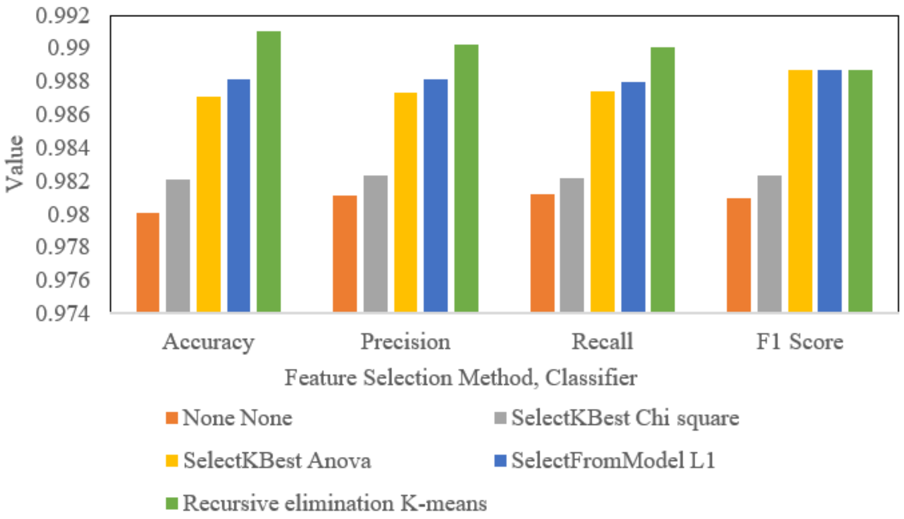


Figure 8. Assessment of FSRE-K-means-CNN Performance Metrics Employing Different Feature Selection Approaches

Examining the figure, it's evident that FSRE-K-means consistently outperforms the other feature selection techniques, albeit with a slight edge over the K-Anova method. The exceptional performance of FSRE-K-means can be attributed to its adeptness at effectively filtering out less informative features, particularly in scenarios with limited sample sizes. In contrast, the performance metrics of the FSRE-K-means-CNN method without employing any feature selection, result in comparatively lower values. Table 3 furnishes a quantitative summary of the performance metrics for various feature selection methods. The initial row presents the performance metrics without the application of any feature selection technique, while the subsequent rows detail the performance metrics for the respective feature selection methods. The "Recursive Elimination" method combined with the "K-means" classifier achieved the highest accuracy and precision, although it had slightly lower recall and F1 Score compared to other methods. "SelectFromModel" with the "L1" classifier and "SelectKBest" with "Anova" also demonstrated strong performance, maintaining high accuracy and precision, with recall and F1 Score scores like the top-performing method. "SelectKBest" with the "Chi Square" classifier and the "None" method with "None" (no feature selection) showed slightly lower performance across all metrics.

Table 3. Performance Analysis of FSRE-K-means-CNN using various Feature Selection Methods

Feature Selection Method	Classifier	Accuracy	Precision	Recall	F1 Score
Recursive elimination	K-means	0.9910	0.9902	0.9901	0.9887
SelectFromModel	L1	0.9881	0.9881	0.9880	0.9887
SelectKBest	Anova	0.9871	0.9873	0.9874	0.9887
SelectKBest	Chi square	0.9821	0.9823	0.9822	0.9823
None	None	0.9801	0.9811	0.9812	0.9810

Moreover, to optimize performance, we experimented with different architectural configurations of the convolutional neural network by adjusting the multipliers. The goal was to obtain various feature map sizes relative to the base feature map size (64, 32, 16). A multiplier of 0.5, for example, would yield feature map sizes of (32, 16, and 4). The results of these experiments are summarized in the table below (Table 4).

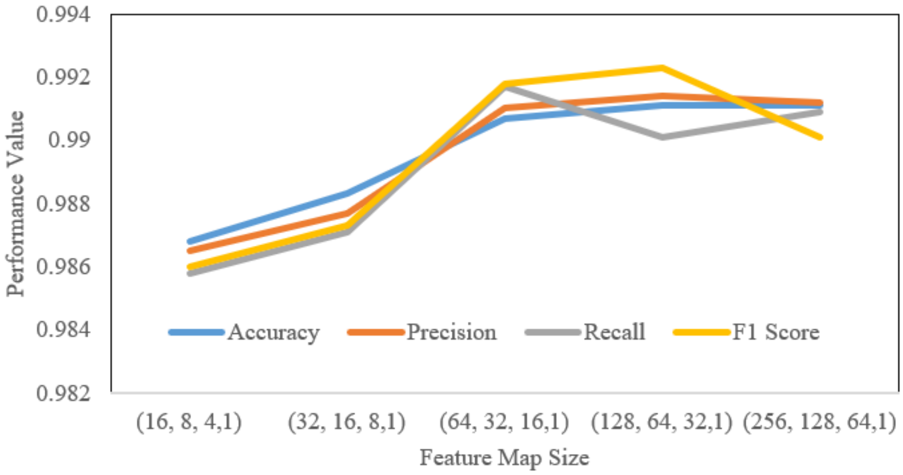


Figure 9. Explore a range of feature map sizes and assess their performance by measuring relevant metrics

As indicated by the results presented in Table 5, the optimal feature map size for the FSRE-K-means-CNN algorithm was determined to be (128, 64, 32, 1). The comparison reveals that larger feature map sizes tend to improve precision and F1 Score, while smaller feature map sizes may impact recall and F1 Score to a certain extent as shown in Figure 9. Selecting the appropriate feature map size depends on the specific requirements of the application, considering the trade-off between model complexity and performance metrics.

Table 5. Experiment with various feature map sizes and evaluate their performance using metrics.

Feature Map Size	Accuracy	Precision	Recall	F1 Score
(256, 128, 64,1)	0.9911	0.9912	0.9909	0.9901
(128, 64, 32,1)	0.9911	0.9914	0.9901	0.9923
(64, 32, 16,1)	0.9907	0.9910	0.9917	0.9918
(32, 16, 8,1)	0.9883	0.9877	0.9871	0.9873
(16, 8, 4,1)	0.9868	0.9865	0.9858	0.9860

We conduct a thorough comparative analysis of FSRE-K-means-CNN in terms of its performance metrics when benchmarked against four existing methodologies ([1, 32, 37, 51, 52]). Impressively, FSRE-K-means-CNN consistently outperforms these methods, achieving the highest levels of accuracy, recall, and F1 scores. To briefly summarize the existing methodologies: [53] employed multi-headed self-attention feature selection combined with a CNN classifier, [54] relied solely on a CNN approach, employed a Random Forest (RF) classifier, [55] utilized a Decision Tree (DT) approach, and made use of Long Short-Term Memory (LSTM). In FSRE-K-means-CNN, we emphasize the pivotal role of FSRE-K-means as a pre-processing step that greatly enhances feature extraction before the subsequent classification phase. Figure 10 visually presents the performance metric results for these six methods, showcasing FSRE-K-means-CNN's superior accuracy, recall, and F1 score. FSRE-K-means-CNN's remarkable performance can be attributed to its multi-step approach, encompassing preprocessing, feature matrix extraction, feature learning, and classification. Additionally, the incorporation of FSRE-K-means significantly contributes to the selection of essential dataset features, resulting in an overall performance boost. Notably, these computationally intensive steps are efficiently managed by the SDN Controller, relieving users' devices of this burden. Furthermore, Figure 10 illustrates the performance metrics which yield results comparable to FSRE-K-means-CNN. Both methods employ similar CNN-based procedures without feature selection. Bahnsen et al. [56] also showcase robust results, primarily attributable to their use of the LSTM method.

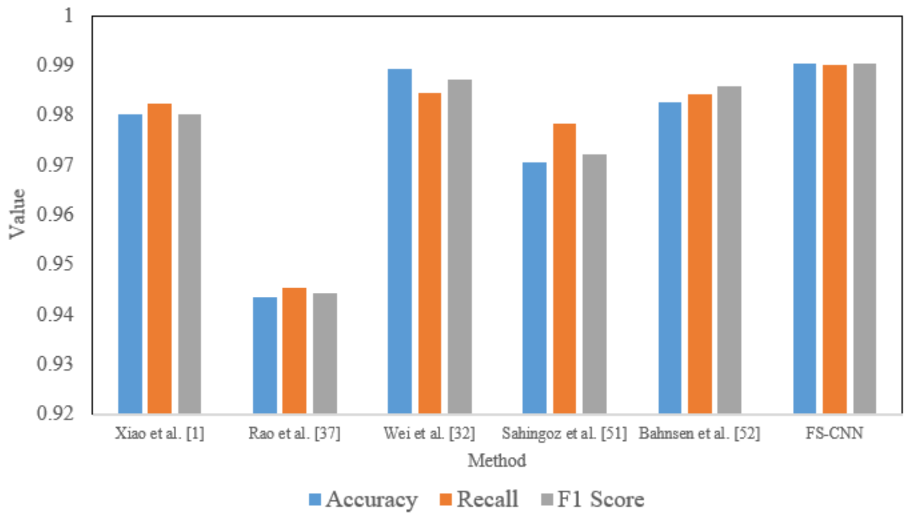


Figure 10. Assessing FSRE-K-MEANS-CNN in Comparison to Benchmarking Methods and Alternative Approaches

Conversely, it indicates that performs less effectively compared to other methods, primarily due to the absence of performance-enhancing techniques or input processing improvements. Additionally, the use of RF classification, while fast and efficient for classification, falls short of the efficiency achieved by CNN on larger datasets. Table 6 provides further details on the performance comparisons among these methodologies.

Table 6. Assessing FSRE-K-means-CNN in Comparison to Benchmarking Methods and Alternative Approaches

Method	**Accuracy**	**Recall**	**F1 Score**
Bahnsen et al. [55]	0.9825	0.9842	0.9858
Sahingoz et al. [38]	0.9704	0.9781	0.9720
Wei et al. [54]	0.9891	0.9843	0.9871
Rao et al. [34]	0.9434	0.9453	0.9442
Xiao et al. [53]	0.9801	0.9823	0.9801
FSRE-K-means-CNN	0.9903	0.9902	0.9904

Bahnsen's method achieved an accuracy of 0.9825, a recall of 0.9842, and an F1 Score of 0.9858, demonstrating a strong overall performance with high accuracy and recall. Sahingoz's method exhibited an accuracy of 0.9704, a recall of 0.9781, and an F1 Score of 0.9720. While its accuracy was slightly lower compared to Bahnsen, it maintained competitive recall and F1 Score. Wei's method showed a high accuracy of 0.9891, a recall of 0.9843, and an F1 Score of 0.9871, excelling in accuracy and F1 Score, though the recall was slightly below Bahnsen. Rao's method reported an accuracy of 0.9434, a recall of 0.9453, and an F1 Score of 0.9442, exhibiting lower performance in all metrics compared to the other methods. Xiao's method delivered an accuracy of 0.9801, a recall of 0.9823, and an F1 Score of 0.9801, showing competitive results, especially in accuracy and recall, like Bahnsen. FSRE-K-means-CNN outperformed all other methods with an accuracy of 0.9903, a recall of 0.9902, and an impressive F1 Score of 0.9904, showcasing the highest performance in accuracy and F1 Score among all the methods. The training procedure for the model involved the utilization of the cross-entropy loss function in combination with the Adam optimization algorithm. To initiate the training, an initial learning rate of 0.001 was employed, and this process extended over 200 epochs. The network's convergence can be clearly observed by referring to the accuracy and loss curves depicted in Figure 11 and Figure 12.

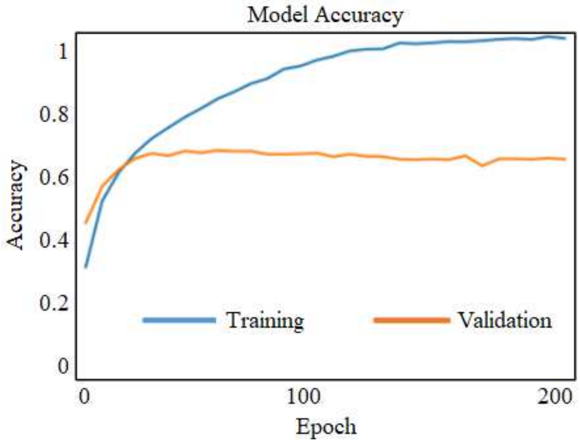


Figure 11. Model Accuracy

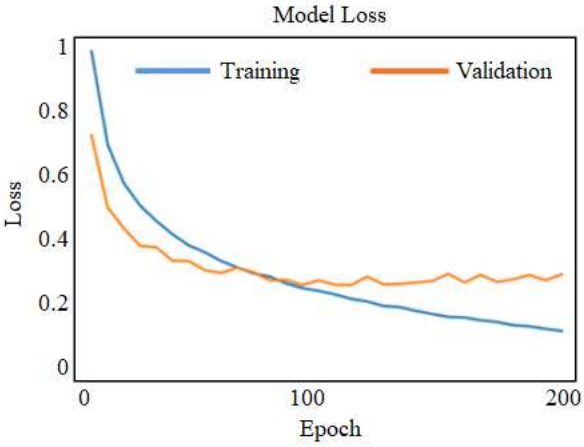


Figure 12. Model Loss

Traditionally, phishing detection solutions have relied on inline inspection methods, such as Intrusion Prevention Systems (IPS) or proxy services, with static string matching akin to traditional Intrusion Detection Systems (IDS) like SNORT [53]. However, deploying these techniques in real-world applications often encounters challenges, including lengthy phishing detection times and static rules that cannot adapt dynamically to real-time phishing threats. Our work seeks to shift these computational demands to SDN, distributing the workload between the SD-Controller and SD-Switches, ultimately aiming to enhance and optimize these processes. In our research on Enhancing Phishing URL Detection Accuracy in Software-Defined Networks (SDNs) through Feature Selection and Adaptive Machine Learning Techniques, we observed a significant increase in the precision of phishing URL identification. Leveraging feature selection allowed us to focus on the most relevant attributes, reducing noise in the dataset and enhancing the overall detection performance. Moreover, the adaptive machine learning techniques exhibited promising capabilities in adapting to evolving phishing tactics, underscoring their potential to bolster the security of SDNs against a dynamic threat landscape. These combined approaches represent a valuable step toward more effective and resilient cybersecurity in SDNs.

Conclusion

Phishing attacks on websites persist as a substantial menace to internet users, and their occurrence has been steadily increasing. To effectively combat this evolving challenge, it is imperative to enhance anti-phishing software. Although machine learning algorithms have become commonplace in this field, the advent of the Internet of Things (IoT) necessitates the adaptation and evolution of these techniques to remain aligned with emerging technologies. In our research, we have introduced a novel methodology that integrates Feature Selection by Recursive Elimination K-means (FSRE-K-means) and Convolutional Neural Network (CNN) methodologies within a comprehensive sequential feature framework. This approach coined FSRE-K-means-CNN, is designed to address the intricacies of phishing detection. It encompasses various stages, including preprocessing, feature matrix extraction, feature learning, and classification. Notably, FSRE-K-means-CNN is tailored for operation within a Software-Defined Networking (SDN) environment, aligning it with contemporary network infrastructures. FSRE-K-means-CNN harnesses the collaborative power of CNN and FSRE-K-means to establish a resilient anti-phishing solution. It operates seamlessly on the SDN controller, with its findings dispatched to the SD-Switch for a thorough examination of each URL packet request. By leveraging SDN technology, this approach enables perpetual training against emerging phishing threats, maintains current phishing countermeasures within the controller, and relieves users' devices of the training workload. Consequently, users benefit from updates that safeguard against a spectrum of phishing attacks without incurring supplementary expenses. Performance evaluation of FSRE-K-means-CNN is based on metrics such as Accuracy, Precision, Recall, and F1-Score. For the SD-Switch component, the inspection time for URL packets is used as a metric. The simulation environment was set up using Mininet and the ONOS controller. FSRE-K-means-CNN achieved an impressive phishing detection accuracy of 99.03%, surpassing the performance of existing methods.

Declaration

- Financial Support: The authors confirm that we did not receive any financial support from any organization for the research presented in this work.
- Ethical statement: Throughout the development of this paper, we maintained strict adherence to ethical guidelines and practices, thereby safeguarding the integrity and credibility of our research.
- Competing Interest: The corresponding author, on behalf of all authors, declares no competing interest in this research.
- Data availability and access: Publicly available in this link https://5000best.com/websites [48].

References

[1] Lin, S. W., Ying, K. C., Lee, C. Y., & Lee, Z. J., "An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection," Applied Soft Computing, vol. 12, no. 10, pp. 3285-3290, Oct. 2012, doi:10.1016/j. asoc.2012.05.004.

[2] Al-Janabi, M., Quincey, E. D., & Andras, P., "Using supervised machine learning algorithms to detect suspicious URLs in online social networks," In Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 1104-1111, July 2017, doi:10.1145/ 3110025.3116201.

[3] Xiao, X., Zhang, D., Hu, G., Jiang, Y., & Xia, S., "CNN–MHSA: A Convolutional Neural Network and multi-head self-attention combined approach for detecting phishing websites," Neural Networks, vol. 125, pp. 303-312, May 2020, doi:10.1016/j. neunet.2020.02.013.

[4] Lu, X., Han, D., Duan, L., & Tian, Q., "Intrusion detection of wireless sensor networks based on IPSO algorithm and BP neural network," International Journal of Computational Science and Engineering, vol. 22, no. 2-3, pp. 221-232, 2020, doi:10.1504/IJCSE.2020.107344.

[5] Rustam, Z., & Kharis, S. A. A., "Comparison of support vector machine recursive feature elimination and kernel function as feature selection using support vector machine for lung cancer classification," In Journal of Physics: Conference Series, vol. 1442, no. 1, pp. 012027, 2020, doi:10.1088/1742-6596/1442/1/012027.

[6] M. Alweshah, M., Alkhalaileh, S., Albashish, D., Mafarja, M., Bsoul, Q., & Dorgham, O., "A hybrid mine blast algorithm for feature selection problems," Soft Computing, vol. 25, pp. 517-534, Jan. 2021, doi:10.1007/s00500-020-05164-4.

[7] "Feature selection," Scikit-Learn. https://scikit-learn.org/stable/modules/feat ure_selection.html#univariate-feature-selection (accessed Jul. 03, 2021).

[8] V.N. Vapnik, "An overview of statistical learning theory," IEEE Transactions on Neural Networks, vol. 10, no. 5, pp. 988–999, Sep. 1999, doi:10.1109/72.788640.

[9] O.A. Khashan, R. Ahmad, N.M. Khafajah, "An automated lightweight encryption scheme for secure and energy-efficient communication in wireless sensor networks," Ad Hoc Networks, vol. 115, pp. 102448, Apr. 2021, doi:10.1016/j. adhoc.2021.102448.

[10] R. Wazirali, R. Ahmad, A. Al-Amayreh, M. Al-Madi, A. Khalifeh, "Secure watermarking schemes and their approaches in the iot technology: an overview," Electronics, vol. 10, no. 14, pp. 1744, Jul. 2021, doi:10.3390/ electronics10141744.

[11] "Software-Defined Networking (SDN) Definition," ONF. https://www.openn etworking.org/sdn-definition (accessed Jun. 08, 2021).

[12] T. Chin, K. Xiong, and C. Hu, "Phishlimiter: a phishing detection and mitigation approach using software-defined networking," IEEE Access, vol. 6, pp. 42513–42531, Jun. 2018, doi:10.1109/ACCESS.2018.2837889.

[13] K. Archana Janani, V. Vetriselvi, R. Parthasarathi, G., and Subrahmanya VRK Rao, "An Approach to URL Filtering in SDN," International Conference on Computer Networks and Communication Technologies, Springer Singapore, pp. 217–228, 2019, doi:10.1007/978-981-10-8681-6_21.

[14] Y. Zhang, J.I. Hong, and L.F. Cranor, "Cantina: a content-based approach to detecting phishing websites," Proceedings of the 16th International World Wide Web Conference WWW2007, pp. 639–648, doi:10.1145/1242572.1242659.

[15] S. Ustebay, Z. Turgut, and M.A. Aydin, "Intrusion Detection System with Recursive Feature Elimination by Using Random Forest and Deep Learning Classifier," International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism, pp. 71–76, 2019, doi:10.1109/IBIGDELFT.2018.8625318.

[16] R. Wazirali, and R. Ahmad, "Machine learning approaches to detect DoS and their effect on WSNs lifetime," Computers. Materials & Continua, vol. 70, no. 3, pp. 4921–4946, Mar. 2022, doi:10.32604/cmc.2022.020044.

[17] Vinayakumar R, Alazab M, Soman KP, Poornachandran P, Al-Nemrat A, and Venkatraman S, "Deep learning approach for intelligent intrusion detection system," IEEE Access, vol. 7, pp. 41525–41550, Apr. 2019, doi:10.1109/ACCESS.2019.2895334.

[18] W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, and M. Wo´zniak, "Accurate and fast URL phishing detector: a convolutional neural network approach," Computer Networks, vol. 178, pp. 107275, Sep. 2020, doi:10.1016/j.comnet.2020.107275.

[19] M. Miao, and B. Wu, "A flexible phishing detection approach based on software-defined networking using ensemble learning method," ACM International Conference Proceedings Series, pp. 70–73, Jun. 2020, doi:10.1145/3407947.3407952.

[20] Abdullaziz OI, and Wang L, "Mitigating DoS Attacks against SDN controller using information hiding," In: 2019 IEEE Wireless Communications and Networking Conference (WCNC), pp 1-6, 2019, doi:10.1109/WCNC.2019.8885764.

[21] Agborubere B, and Sanchez-Velazquez E, "OpenFlow communications and TLS security in software-defined networks," In: 2017 IEEE International Conference on Internet of Things (IThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp 560–566, doi:10.1109/iThings-GreenCom-CPSCom-SmartData.2017.88.

[22] Ahmad I, Namal S, Ylianttila M, and Gurtov A, "Security in software defined networks: a survey," IEEE Communications Survey & Tutorials, vol. 17, no. 4, pp. 2317–2346, Aug. 2015, doi:10.1109/COMST.2015. 2474118.

[23] Ahmed ME, and Kim H, "DDoS attack mitigation in Internet of things using software-defined networking," In: 2017 IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService), pp 271–276, Apr. 2017, doi:10.1109/BigDataService.2017.41

[24] Aizuddin AA, Atan M, Norulazmi M, Noor MM, Akimi S and Abidin Z, "DNS Amplification attack detection and mitigation via sFlow with Security-Centric SDN," In: Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication, pp. 1-7, Jan. 2017, doi:10.1145/3022227.3022230.

[25] Al-Haj S, and Tolone WJ, "Flow Table pipeline misconfigurations in Software Defined Networks," In: 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp 247–252, May 2017, doi:10.1109/INFCOMW.2017.8116384.

[26] Alasadi E, and Al-Raweshidy HS, "SSED: servers under software-defined network architectures to eliminate discovery messages," IEEE/ACM Transactions on Networking, vol. 26, no. 1, pp. 104–117, Nov. 2018, doi:10.1109/TNET.2017.2763131.

[27] Alcorn JA, and Chow CE, "A framework for large-scale modeling and simulation of attacks on an OpenFlow network," In 2014 23rd International Conference on Computer Communication and Networks (ICCCN), pp. 1–6, Aug. 2014, doi:10.1109/ICCCN.2014.6911848.

[28] Allouzi M, and Khan J, "SafeFlow: authentication protocol for software-defined networks," In: 2018 IEEE 12th International Conference on Semantic Computing (ICSC), pp 374–376, Jan. 2018, doi: org/10.1109/ICSC.2018.00076.

[29] Alparslan O, Gunes O, Hanay YS, Arakawa S, and Murata M, "Improving resiliency against DDoS attacks by SDNand multipath orchestration of VNF services," In 2017 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), pp. 1–3, Jan. 2017, doi:10.1109/LANMAN.2017.7972158.

[30] Ambrosin M, Conti M, Gaspari FD, and Poovendran R, "LineSwitch: tackling control plane saturation attacks in software-defined networking," IEEE/ACM Transactions on Networks, vol.25, no.2, pp. 1206–1219, 2017, doi:10.1109/TNET.2016.2626287.

[31] AseeriA, Netjinda N, and Hewett R, "Alleviating eavesdropping attacks in software-defined networking data plane," In: Proceedings of the 12th Annual Conference on Cyber and Information Security Research, pp. 1-8, Apr. 2017, doi:10.1145/3064814.3064832.

[32] De Assis MVO, Hamamoto AH, Abrão T, and Proença ML, “A game theoretical based system using holt-winters and genetic algorithm with fuzzy logic for DoS/DDoS mitigation on SDN networks,” IEEE Access, vol. 5, pp. 9485–9496, May 2017, doi:10.1109/ACCESS.2017.2702341.

[33] P. Yang, G. Zhao, and P. Zeng, “Phishing website detection based on multidimensional features driven by deep learning,” IEEE Access, vol. 7, pp. 15196–15209, Jan. 2019, doi:10.1109/ACCESS.2019.2892066.

[34] R.S. Rao, T. Vaishnavi, and A.R. Pais, “CatchPhish: detection of phishing websites by inspecting URLs, Journal of Ambient Intelligent Humanized Computing, vol.11, no. 2 pp. 813-825, Feb. 2020, doi:10.1007/s12652-019-01311-4.

[35] K.L. Chiew, C.L. Tan, K.S. Wong, K.S.C. Yong, and W.K. Tiong, “A new hybrid ensemble feature selection framework for machine learning-based phishing detection system,” Information Sciences, vol. 484, pp. 153-166, May 2019, doi:10.1016/j.ins.2019.01.064.

[36] E. Zhu, Y. Chen, C. Ye, X. Li, and F. Liu, “OFS-NN: an effective phishing websites detection model based on optimal feature selection and neural network,” IEEE Access, vol. 7, pp. 73271-73284, Jun. 2019, doi:10.1109/ACCESS.2019.2920655.

[37] Mao J, Bian J, Tian W, Zhu S, Wei T, Li A, and Liang Z, “Detecting phishing websites via aggregation analysis of page layouts,” Procedia Computer Science, vol. 129, pp. 224-230, Jan. 2018, doi:10.1016/j.procs.2018.03.053.

[38] O.K. Sahingoz, E. Buber, O. Demir, and B. Diri, “Machine learning based phishing detection from URLs,” Expert Systems with Applications, vol. 117, pp. 345–357, Mar. 2019, doi:10.1016/j.eswa.2018.09.029.

[39] A.C. Bahnsen, E.C. Bohorquez, S. Villegas, J. Vargas, and F.A. Gonzalez, “Classifying phishing URLs using recurrent neural networks,” In 2017 APWG symposium on electronic crime research (eCrime), pp. 1-8, Apr. 2017, doi:10.1109/ECRIME.2017.7945048.

[40] Ruby, A. U., Chandran, J. G. C., Jain, T. S., Chaithanya, B. N., & Patil, R., “RFFE–Random Forest Fuzzy Entropy for the classification of Diabetes Mellitus,” AIMS Public Health, vol. 10, no. 2, pp. 422-442, 2023, doi:10.3934/publichealth.2023030.

[41] Li, Y., Xiao, J., Chen, Y., & Jiao, L., “Evolving deep convolutional neural networks by quantum behaved particle swarm optimization with binary encoding for image classification,” Neurocomputing, vol. 362, pp. 156-165, Oct. 2019, doi:10.1016/j.neucom.2019.07.026.

[42] Islam, M. J., Ahmad, S., Haque, F., Reaz, M. B. I., Bhuiyan, M. A. S., & Islam, M. R., “Application of min-max normalization on subject-invariant EMG pattern recognition,” IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1-12, Nov. 2022, doi:10.1109/TIM.2022.3220286.

[43] Yan, K., & Zhang, D., “Feature selection and analysis on correlated gas sensor data with recursive feature elimination,” Sensors and Actuators B: Chemical, vol. 212, pp. 353-363, Jul. 2015, doi:10.1016/j.snb.2015.02.025.

[44] Likas, A., Vlassis, N., & Verbeek, J. J., “The global k-means clustering algorithm,” Pattern Recognition, vol. 36, no. 2, pp. 451-461, Feb. 2003, doi:10.1016/S0031-3203(02)00060-2.

[45] Aljofey, A., Jiang, Q., Qu, Q., Huang, M., & Niyigena, J. P., “An effective phishing detection model based on character level convolutional neural network from URL,” Electronics, vol. 9, no. 9, pp. 1514, Sep. 2020, doi:10.3390/electronics9091514.

[46] Eckle, K., & Schmidt-Hieber, J., “A comparison of deep networks with ReLU activation function and linear spline-type methods,” Neural Networks, vol. 110, pp. 232-242, Feb. 2019, doi:10.1016/j.neunet.2018.11.005.

[47] G. Çınarer, B.G. Emiro˘glu, R.S. Arslan, A.H. Yurttakal, “Brain tumor classification using deep neural network,” Advances in Science, Technology and Engineering Systems, vol. 5, no. 5, pp. 765–769, Oct. 2020, doi:10.25046/AJ050593.

[48] “5000 BEST WEBSITES.” http://5000best.com/websites (accessed May 21, 2021).

[49] “Open Network Operating System (ONOS),” onosproject. https://docs.onosproject.org/ (accessed Jun. 07, 2021).

[50] “Mininet.” http://mininet.org/ (accessed Jul. 11, 2021).

[51] “Feature selection,” Scikit-Learn. https://scikit-learn.org/stable/modules/feature_selection.html#univariate-feature-selection (accessed Jul. 03, 2021).

[52] “sklearn.feature_selection.SelectFromModel,” Scikit-Learn. https: // scikit-learn.org /stable/ modules/ generated/ sklearn.feature_selection.SelectFromModel.html (accessed Jul. 03, 2021).

[53] X. Xiao, D. Zhang, G. Hu, Y. Jiang, S. Xia, “CNN–MHSA: a convolutional neural network and multi-head self-attention combined approach for detecting phishing websites,” Neural Networks, vol. 125, pp. 303–312, May 2020, doi:10.1016/j.neunet.2020.02.013.

[54] W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, M. Wo´zniak, “Accurate and fast URL phishing detector: a convolutional neural network approach,” Computer Networks, vol. 178, Apr. 2020, doi:10.1016/j.comnet.2020.107275.

[55] O.K. Sahingoz, E. Buber, O. Demir, B. Diri, “Machine learning based phishing detection from URLs,” Expert Systems with Applications, vol. 117, pp. 345–357, Mar. 2019, doi:10.1016/j.eswa.2018.09.029.

[56] A.C. Bahnsen, E.C. Bohorquez, S. Villegas, J. Vargas, F.A. Gonzalez, “Classifying phishing URLs using recurrent neural networks,” In 2017 APWG symposium on electronic crime research (eCrime) IEEE, pp. 1–8, Apr. 2017, doi:10.1109/ECRIME.2017.7945048.