International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6s
DOI: https://doi.org/10.17762/ijritcc.v11i6s.6957
***Article Received**: 26 March 2023 **Revised**: 18 May 2023 **Accepted**: 01 June 2023*

Malicious URL Website Detection using Selective Hyper Feature Link Stability based on Soft-Max Deep Featured Convolution Neural Network

M. Pushpalatha[1], Dr.A. Vijaya[2]
[1]Ph.D. Research Scholar (PT), Assistant Professor, Department of Computer Science,
Padmavani Arts & Science College for Women,
Salem, TamilNadu.India - 636011
lathapmj79@gmail.com
[2]Research Supervisor, Assistant Professor & Head, Department of Computer Applications,
Govt. Arts College (Autonomous),
Salem-636 007, TamilNadu. India
vijayakathiravan@gmail.com

Abstract- The web resource contains many domains with different users' Uniform Resource Locators (URLs). Due to the increasing amount of information on the Internet resource, malicious activities are done by hackers by expecting malicious websites in URL sub-links. Increasing information theft leads data sources to be vested in huge mediums. So, to analyze the web features to find the malicious webpage based on the deep learning approach, we propose a Selective Hyper Feature Link stability rate (SHFLSR) based on Soft-max Deep featured convolution neural network (SmDFCNN) for identifying the malicious website detection depends on the actions performed and its feature responses. Initially, the URL Signature Frame rate (USFR) is estimated to verify the domain-specific hosting. Then the link stability was confirmed by post-response rate using HyperLink stability post-response state (LSPRS). Depending upon the Spectral successive Domain propagation rate (S^2DPR), the features were selected and trained with a deep neural classifier with a logically defined Softmax- Logical activator (SmLA) using Deep featured Convolution neural network (DFCNN). The proposed system performs a high-performance rate by detecting the malicious URL based on the behavioral response of the domain. It increases the detection rate, prediction rate, and classifier performance.

Keywords: Malicious website Deep learning; Convolution neural network; feature selection; Link stability; Domain subsequent rate analysis; Hyper feature selection.

I. INTRODUCTION

The impact of internet technology on web resources has grown considerably and comes under various threats. All over the web resource are carried out through Uniform Resource Locator (URL), which is vulnerable to attacks such as phishing and spam. To protect yourself from these attacks, it is essential to inquire about malicious URLs. However, there are still issues with current studies. The URL links are responded to base on the post back domain, which is handled by anomaly interacted features some methods can be easily bypassed by attackers. Machine learning algorithms are typically used to construct a deep neural network (DNN) neural architecture by detecting malicious URLs and predicting feature mapping concerning support vector values to reduce dimensionality. Additionally, the limits of the pooling layer are accustomed to competition with the length of the URL entry and the depth of the current convolution layer. It allows malicious features to extract detailed features over a wide range. In this research, the optimized URL detection be carried through deep neural feature selection and classification model—the DNN method for learning the vector representations of URLs using word embedding based on written embedding.

In the meantime, we will conduct a comparative experiment between the two groups. To perform different tests using the same network system and other embedding methods. This result shows that greater accuracy can be achieved by supporting links by verifying domain categories. By improving the classification accuracy, the web resources selectively analyze the feature based on the feature selection approach. The importance of hyperlinks is verified with domain-related links features, and the stability rate was estimated. URLs act as a linking process to create identities and signatures to verify a user's access to a domain. Through analysis, malicious actors are analyzed by the rate of behavior accessed by queries handled by the post-response rate, which redirects the page link to the correct domain. Otherwise, the feature access will be malicious. So creating false positive and negative measures to achieve through domain generation algorithms. Based on the success rate of domain support

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6s
DOI: https://doi.org/10.17762/ijritcc.v11i6s.6957
***Article Received**: 26 March 2023 **Revised**: 18 May 2023 **Accepted**: 01 June 2023*

values to find the malicious URLs. Researchers use machine-learning techniques to detect malicious URLs. However, these methods often require manual extraction of features designed and undetected by the attacker. In today's complex network environment, research focuses on creating the most effective malicious URL detection models.

This paper proposes a DFCNN-based malicious URL detection model. It accepts spelling-based word embedding, automatically extracts features, and learns URL expressions. On the other hand, we will check the model's validity through a series of comparative tests.

II. RELATED WORK

In recent years, detecting malicious URLs directly has become a new research direction. The features utilize character embedding technology to convert the original URL into a feature vector after transferring it into the malicious software URL detection system [1] and training using a Convolutional Neural Network (CNN) [2]. It significantly reduces the dimensions and complexity of the data. Most analysts analyze the structure and function of in-depth learning of different URLs, extract the function, and propose a semi-supervisory training model for multiple classifications of URLs [3]. It extracts the functionality of the URL domain name and the temporary function of the redirect attack, improves the neural network structure, and improves detection accuracy.Next, we will introduce an original detection method that takes screenshots of malicious web pages that take the user's perspective and disable internet spam [4]. Convolutional Neural Networks, a subset of deep learning, are used as a classification technique in the suggested detection method. Two independent tests were undertaken to validate the usefulness of this strategy [5]. First, test the proposed method on a database with a complicated structure. Early detection systems aren't based on machine learning techniques. Propose a blockchain system that analyzes malicious website domain information, evaluates Internet Protocol (IP) address blocks and registrar reputations, and on this basis, associates IP address blocks with less reputable registrars [6]. Ranking schemes use the attacker's history and current log generation methods to measure the proximity of attack sources and contributors [7]. Block list systems combine intensity signs with log previews to determine how well an attacker's alert system corresponds to popular malware distributions [8]. Deep learning methods are being used by more researchers today to identify malicious websites. Deep learning and conventional machine learning methods [9].

Artificial Neural Networks (ANN) work best by using SVM, Decision Tree (DT), Naive Bayes (NB), and KNN and as selected URL phrase and page content features and classification algorithms [10]. Deep Learning (DL) methods are generally better than traditional methods. However, those databases balance the number of malicious and legitimate URLs [11]. It does not happen on the real-world Internet. In the real world of the Internet, positive and negative imbalanced data can make detection difficult and reduce detection accuracy [12]. Also, spamming strategies that slow detection performance is not considered.Each study examines the neural network configuration, parameter optimization, and performance measurements to gain a complete understanding of the design [13], implementation, and evaluation of each DL model. Finally, compared to other research in the same field, the novelty of our work is highlighted [14]. However, a significant amount of labeled data is necessary for DNN training [15].Furthermore, parameter selection techniques must be optimized, and the learning process is time-consuming [16]. Despite these shortcomings, many studies have been conducted to investigate the effectiveness of using DNNs to detect phishing web pages [17].

Moreover, performance metrics are another important component used to study and evaluate phishing detection systems. A previous study showed that the performance of DL models in detecting phishing networks was assessed using a limited number of measurements [18]. The difference between CNN and DNN frameworks is the usage of convolutional layers and kernels. Recognizing the important role of these factors in determining the performance accuracy of phishing detection models, most researchers consider learning rate, dropout rate, epochs, and block size to be more important than other parameters. After that, we can focus on specifying parameters [20] has been resolved, but the program should provide mechanisms to improve these parameters. Likewise, the authors describe although this is an optimization procedure, only some parameters are of interest, such as the number of convolutional layers [21] kernels, and kernel size.

III. PROPOSED SOLUTION

Towards the malicious website detection, the proposed system integrates a Selective Hyper Feature Link stability rate based on Soft-max Deep featured convolution neural network for identifying the malicious website detection depending on the actions performed and its feature responses. These features are extracted from the URL collection in the feature extraction and representation phase. We train a classification model by combining the extracted features with vectors directly processed by a convolutional neural network.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6s
DOI: https://doi.org/10.17762/ijritcc.v11i6s.6957
***Article Received**: 26 March 2023 **Revised**: 18 May 2023 **Accepted**: 01 June 2023*

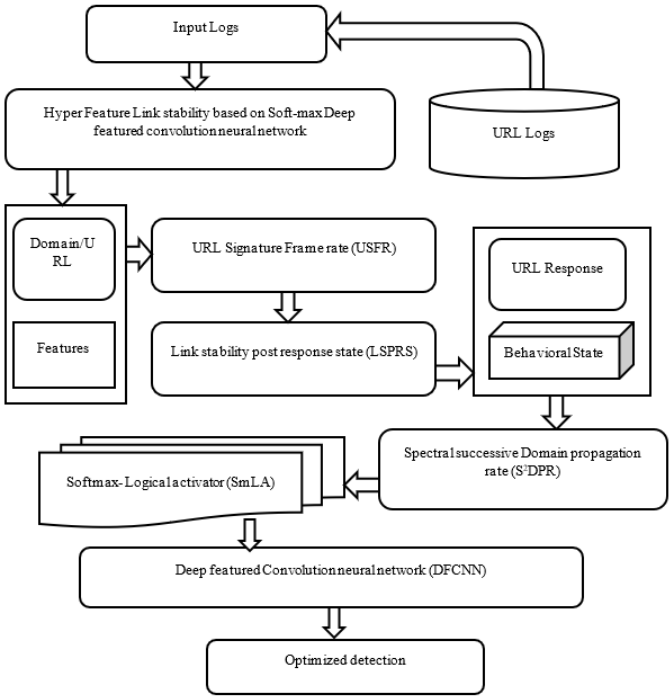

Figure 1. Proposed architecture diagram SHFLSR-SmDFCNN

This method simplifies the feature extraction process, does not rely on manual feature extraction, and combines the benefits of semantic and word insertion. Figure 1 Proposed architecture diagram SHFLSR- SmDFCNN. Link embedding can retrieve word sequence information that cannot be recovered. Handles special characters and embeds unfamiliar words in the URL. Both dictionaries and vectors have very small dimensions. This combination saves memory space and allows URLs to be represented more efficiently to predict the specific domain. It enables the extraction of information from the URL. The following is a detailed proposed step and detection process for each module.

A. URL Signature Frame rate (USFR)

This position attempts to extract features by analyzing URLs and related website or webpage information. The features extracted in this way are generally divided into two categories: static features and dynamic features. The URL signatures dependably relate to base physical layer protocols used in malicious websites. Each Link post be verified by access link through digital signing certificates and secure socket Layer (SSL) certificates related to hosting domains. During the domain specifications, the URL request is verified.

A URL is classified as malicious if it matches malicious signature data. The form below represents the data verified by the URL signature vector and frequency.

$$(V_{ec}, Fr)= (p_1, p_2, p_3, .. p_n, F) \qquad (1)$$

The vector (V_{ec}) is the combined set of maliciousp_1, p_2, p_3 and p3, where F is the dependency (output) that represents the Classification.

Let $U_n$be URL for Analysis. A set of malicious URLs in configuration files $p_1, p_2, p_3, .. p_n$

$$\text{If } U_n \in \{U_i\} \text{ where i=1, 2…n}$$

Compute the sum of frequencies of s_m across all begin files f_i and normalize it to $F(s_m)$

$$F(s_m) = (\sum_n^i f_i(s_m))/N_i \qquad (2)$$

Divide the sum of frequencies $s_m$across all malware files and normalize it to$F(M_a)$

$$F(M_a) = s_m(\sum_n^i f_i(M_a))/N_a \qquad (3)$$

Find each difference in normalized frequency at signature$D(F)$

$D(F) = |F_s(D(F)) - F_M((F))|$ Be return Stability support 'S.'

The three main modules that make up a convolutional neural network based malicious URL detection model are the vector embedding module, dynamic convolution module and block extraction module.

B. Link stability post response state (LSPRS)

Link stability proves the reliable nature of redirection URLs to find the exact post-processing after getting a good link. This stability increases the URL reliability rate for redirecting the page to present hyperlinks domains. By identifying the strong support, calculate the average means rate by the support and confidence measure.

The equation below calculates URL link stability support (S) information.

$$\text{Support (S)} = \frac{\sum_j^{j=1} \sum_k^{k=1} S_c}{\sum T_t} \qquad (4)$$

Let's assume j refers to the row, k refers to the column, S_c denotes URL support count, and T_t refers to the number of redirect link processes. To analyze, the URL confidence state (C_u) performance can express as,

$$C_u = \frac{\sum p \cup N}{\sum S_c} \qquad (5)$$

Let's assume p refers to the Predecessor of the URL dataset, and N refers to the importance of the URL. The equation is used to analyze website URL support information. The redirection of the domain link has the maximum support to get the specific values from the redirection page of the URL in the main domain.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6s
DOI: https://doi.org/10.17762/ijritcc.v11i6s.6957
***Article Received**: 26 March 2023 **Revised**: 18 May 2023 **Accepted**: 01 June 2023*

C. Spectral successive Domain propagation rate (S^2DPR)

The features get extracted to select the importance of the domain-specific within the relative point domain (URLs) supports, which is actively successful to the point in support values. The malicious URL feature representation highlights key features in detecting malicious URL features. It represents the URL feature vector (the URL represents the size of the data, the dimensions of the feature vector) and the importance of each feature in the URL. Vector is the average weight of the blocks of data. This feature is very important for malicious URL detection when the data block is large. The input and output values of the eigenvector focusing mechanism are derived from the resolution rate.

Algorithm

Input: Link stability maximum support values (M_S)

Output: Importance of features (f_i)

Begin

Step 1: Initialize (M_S) the URL maximum support values

Step 2: For each j=1 do then

// j refers to the row

For each k=1 do then

// k refers to the column

Compute URL relevance feature vector

$$M_S \in f_m^{iXD}$$

Calculate the importance of the features rate from URL (I_m)

$$I_m = \sum_j^i w_c$$

//w_c represents the URL coefficient weight

To find coefficient weight w_c

$$w_c = \frac{\exp[f(g_i)]}{\sum \exp[f(g_i)]} \quad (6)$$

f denotes function of g_i denotes encoding mechanism

End for

End for

Step 3: Calculate URL maximum successive feature rate

$$f_i = \frac{\sum_{i=1}^{j=1} I_m}{|M_S|} \quad (7)$$

Step 4: Return← $the\ f_i$

Stop

The above algorithm steps efficiently analyze the URL successive feature rate using Spectral successive Domain propagation rate (S^2DPR). M_SMaximum support values of the URL dataset, f_m denotes the URL features matrix, i indicates the number of URL, and D represents the dimension feature vector. The focus model derives the attribute values of the URL. Features that significantly impact the URL classification will be selected and given more weight to participate in the next calculation step. Therefore, the important URL function for detecting malicious URLs can be obtained as input.

D. Softmax- Logical activator (SmLA)

In this stage, logical activation functions are carried out to provide functional strategies within the marginal (Weighted) threshold values. This can be evaluated by the functional approach by predicting the maximum threshold value pointed to the CNN for using the pooling layer by supporting values of URL dependencies. Softmax function creates logical evaluation in neural dependencies having specified pooled layer conditional verification of attaining feature margins holder to weight optimization.

Algorithm

Input: Importance of features (f_i)

Output: Soft-max logical activator (S_l) dataset

Begin

Step 1: Initialize the Importance of features (f_i)

Step 2: For j=1, do then

Compute each neuron soft-max activation function (σ) rule

$$\sigma(f_i)_n = \frac{\exp(f_i)}{\sum_{j=1}^{i} exp\ (f_i)} \quad (8)$$

End for

Step 3: Calculate the loss function

$$L(P(\sigma(f_i)), E(S_l)) = -\frac{1}{n}\sum_{j=1}^{n}\left[E(S_l)_{jk} \log P(\sigma(f_i))_{jk} + \left(1 - E(S_l)_{jk}\right) \log\left(1 - P(\sigma(f_i))_{jk}\right)\right] \quad (9)$$

// $P(\sigma(f_i))$ refers to URL predicted value, $E(S_l)$ refers to URL expected class value

Step 4: Sorting and update the URL weightage based on the soft-max rule

Step 5: Return ← S_l

Stop

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6s
DOI: https://doi.org/10.17762/ijritcc.v11i6s.6957
***Article Received**: 26 March 2023 **Revised**: 18 May 2023 **Accepted**: 01 June 2023*

Let's assume σ denotes the soft-max activation function of input vectorf_i, n refers to the n[th] element of input iteration, $\sum_{j=1}^{i} exp\ (f_i)$ represents the output of the logical activator functionS_l. It selects the marginal weights during Classification to spread the dependencies weights to each pooling layer in neural structures to verify the condition. If the URL margins get conditional substances of behavioral activities and check the match case values if the importance of the feature gets support to sustain during the Classification. It logically improves the conditional verification approach to predict the URL malicious dependencies and improve classification accuracy.

E. Deep-featured Convolution neural network (DFCNN)

The dynamic convolutional method extends the multi-layer convolutional structure by a new convolutional layer. The k-max-pooling layer should take the position of the pooling layer. In the dynamic convolution method, the magnitude of the input vector determines the width of the hidden layer feature map. In addition, we also change the length of the URL input and the depth of the current convolutional layer by adjusting the pooling layer parameters to extract detailed features. An upgraded CNN network is created and taught using the text embedding technique and the transformation.URL group. The pack-proof system continuously updates numerous example parameters while the CNN network is prepared to utilize a URL tutorial. After finishing the example lesson, you will have extracted multi-layered URL features from the CNN network.

Algorithm steps

Input: Selective features (S_l) dataset

Output: Optimization results in Malicious URL website or Not

Begin

Step 1: Initialize the Soft-max logical activator **(S_l)**

$$S_l=\{S_{l1,}S_{l2}, \dots, S_{ln}\}$$

Step 2: For i=1 in m, do

// m refers to the iteration of S_l

Compute hidden layer process H_L

$$H_L = \sigma(QS_l + z \odot H_{T-1} + B) \quad (10)$$

End for

Step 3: Calculate entropy loss function $e(\theta)$

$$e(\theta) = -\frac{1}{n}\left[\sum_{j}^{j=1}\sum_{k}^{k=1}\log(H_L)\right] log\ \left(O^{(i)}, \theta\right)$$

Step 4: Check whether the malicious URL or not using the soft-max activator rule

$$\sigma\left(O^{(i)}, \theta\right) = \frac{\exp\left(\theta_j^i H_L\right)}{\sum_{j=1}^{i} exp\ (\theta_j^i H_L)} \quad (11)\theta$$

refers to classification parameter output O.

Step 5: While ($\sigma\left(O^{(i)}, \theta\right) > \beta$) (12) β refers to a threshold value

Calculate the output layer process

$$O_L = \sigma(\sum H_L\, Q + B) \quad (13)$$

End while

Step 6: Return ← O_L

Stop

The above algorithm steps provide efficient classification results using the proposed DFCNN algorithm. Let's assume Q refers to input weight, u refers to pooling weight, and B refers to bias. ⊙ Presents the product, S_l denotes URL soft-max logical input vector, and T refers to time.

Multi-level features that enhance the precision and generalization of the classification algorithms can be used to classify malicious online content. To extract multi-level URL features and thorough URL feature information from an extended CNN network, we divide high-latitude, mid-latitude, and low-latitude features. It has a significant effect and is frequently applied to increase the effectiveness.

IV. RESULT AND DISCUSSION

The results are computed in a network simulator with an effective database for predicting network traffic using complicated matrix estimations and random data processing. On a wide range of detected malicious activities, the stability of the chosen hyper-functional connections based on Soft-Max Deep Function Convolutional Neural Networks (SmDFCNN) was tested, and the proposed useful classifiers SVM, PDSBF, and CNN produced more effective results than other feature selection classifiers like DS2RNN. The suggested carry forcing has a greater detection rate by sorting the results by their class order. The expanded design of the simulated instrument, which uses a system configuration of 4GB RAM and an i3 Intel processor, is a real-world representation of the issue.

Table 1 shows the details of the malicious website prediction dataset that is processed to test the presence of the planned schemes.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6s
DOI: https://doi.org/10.17762/ijritcc.v11i6s.6957
***Article Received**: 26 March 2023 **Revised**: 18 May 2023 **Accepted**: 01 June 2023*

TABLE I DETAILS OF PARAMETERS

Parameters	Values
Dataset used	KDD-cup domain Urlweb Dataset
environment	Python/Jupiter notebook
Test result	Confusion matrix
Number of features	30
Number of class	3(High medium-low)

Presentation is suppressed based on measures like placement accuracy, sensitivity, specificity levels, time complexity, and misclassification.

$$\text{Accuracy} = \frac{\text{TN+TP}}{(\text{TP+FP+FN+TN})} \quad (14)$$

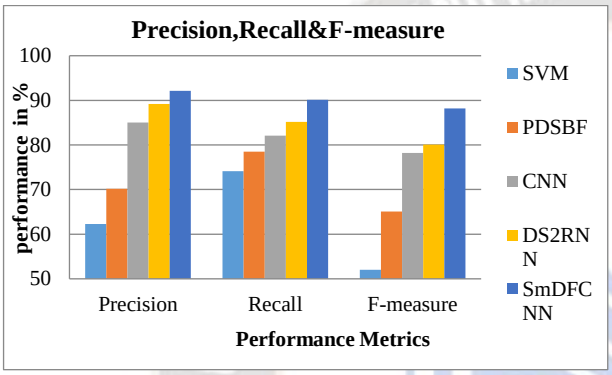

Figure 2. Impact of performance precision-recall and f-measure

Performance defines the precision and uniqueness of frequent measurements predicted by fit/recall generated by positive values and shows absolute results based on type class. Figure 4 shows the performance of the Classification.

TABLE II IMPACT OF PERFORMANCE METRICS

Performance metrics	SVM	PDSBF	CNN	DS2RNN	SmDFCNN
Precision	62.3	70.2	85	89.2	92.1
Recall	74.1	78.5	82.1	85.2	90.1
F-measure	52	65.1	78.2	80.1	88.2

Table 2 shows the classification accuracy compared to the different methods. The proposed method has the best performance in detecting malicious websites. Additional positive correlations are obtained with the true value of the negative range. Classification.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}. \quad (15)$$

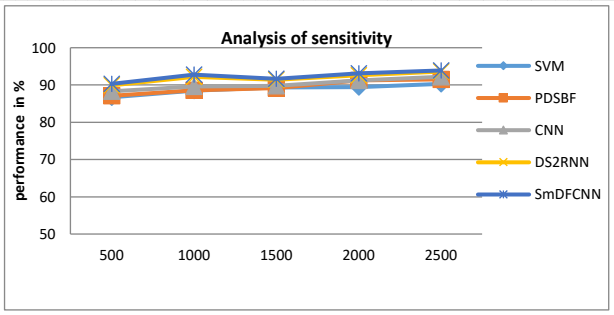

Figure 3 Impact of sensitivity analysis

In the functional assessment of the classification results, Figure 3 defines the level of sensitivity of the detection results for various dataset records. The projected SmDFCNN system outperformed the current ensemble approaches regarding performance rate.

TABLE III IMPACT OF SENSITIVITY ANALYSIS

Methods/No of Records	Impact of Sensitivity Analysis in %				
	SVM	PDSBF	CNN	DS2RNN	SmDFCNN
500	86.7	87.1	88.3	89.9	90.3
1000	88.5	88.6	89.6	92.2	92.8
1500	89.3	89.2	89.8	91.4	91.7
2000	89.4	91.2	91.2	92.6	93.1
2500	90.3	91.5	92.2	93.6	93.9

Table 3 reviews the sensitivity analysis formed and shows the true values of the positive drug test, defined as the true negative divided by the false positive value to adhere to the Classification. The specificity is calculated as follows.

$$\text{Specificity /precision} = \frac{\text{TN}}{\text{TN+FP}} \quad (16)$$

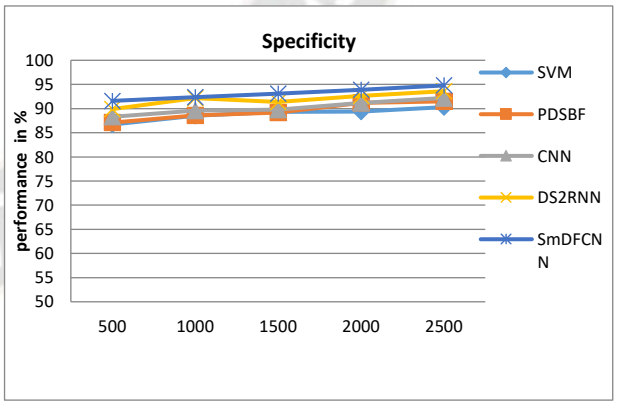

Figure 4 Impact of specificity

Figure 4 shows a variation of the so-called integrity generated by different approaches, where the DS2RNN method produces higher performance than the additional methods.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6s
DOI: https://doi.org/10.17762/ijritcc.v11i6s.6957
***Article Received**: 26 March 2023 **Revised**: 18 May 2023 **Accepted**: 01 June 2023*

TABLE IV IMPACT OF SPECIFICITY

Methods/No of Records	Impact of Specificity in %				
	SVM	PDSBF	CNN	DS^{2}RNN	SmDFCNN
500	82.3	87.3	89.3	91.3	91.6
1000	83.8	87.6	91.2	91.8	92.4
1500	84.2	88.5	92.6	92.8	93.1
2000	85.3	88.9	92.8	93.2	93.9
2500	86.3	90.2	93.5	94.5	94.8

The criterion of accuracy, depending on the percentage of true values in the database, represents the coherent representation that avoids false positives and false negatives. Table IV shows different methods of analyzed accuracy to indicate the ratio. The proposed approach emphasizes higher efficiency compared to other methods.

$$Precision\ values\ are\ calculated\ by\ = \frac{TP}{(TP+FP)};$$ The detection depends on the estimated values. $$Recall\ value\ i\ is\ calculated\ by\ = \frac{TP}{(TP+FN)}$$ By this two-measure, F − measure (False Classification) $=\frac{2*\text{Precision}*\text{Recall}}{(\text{Precision + Recall})}$, by the error rate under 2.5 mean average rates is defined as absolute error 97.93 % well classification accuracy. As followed the confusion matrix shown below

Classifying the original data as the residual form is misclassified to define it as a well-defined class. The expression of this expression is identified as follows:

$$\text{By False Extraction Ratio (Fer)} = \sum_{k=0}^{k=n} \times \frac{TotalDataset\ FailedtoClassify\ (Fer)}{TotalnoofData(Fr)} \quad (17)$$

Figure 5. Impact of false Classification

Figure 5 shows the variance of false ratios generated by different approaches, and the predicted SmDFCNN method has fewer false classifications than the rest of the additional approaches.

TABLE V IMPACT OF FALSE CLASSIFICATION

Methods/No of Records	Comparison of False Classification in %				
	SVM	PDSBF	CNN	DS2RNN	SmDFCNN
500	6.6	6.3	5.9	5.1	4.9
1000	7.2	7.1	6.5	6.3	5.8
1500	8.3	7.8	6.6	6.5	6.1
2000	8.6	7.5	6.8	6.7	6.3
2500	8.8	7.9	6.9	6.8	6.5

Table V bove explains that the misdiagnosis failed Classification is classified as a misclassified class, regardless of whether the data were identified as a trained class or an attack. The misclassification detection class has the lowest predictive value for the proposed SmDFCNNsystem compared to other systems.

$$\text{Time complexity (Tc)} = \sum_{k=0}^{k=n} \times \frac{\text{Total Features Handeled to Process in Dataset}}{\text{Time Taken(Ts)}} \quad (18)$$

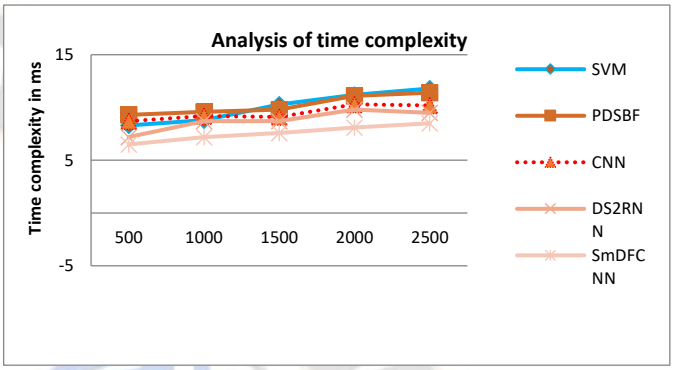

Figure 6. Impact of time complexity

Detection accuracy is designed based on the time it takes to process. Compare detection accuracy in different ways. O (n) will take time for better detection by processing all records based on the type definition of the defined type. The proposed system SmDFCNN produces 9.1 (ms) higher performance than all other previous systems, as shown in Figure 6.

TABLE VI IMPACT OF TIME COMPLEXITY

Methods/No of Records	Impact of Time Complexity in Milliseconds (ms)				
	SVM	PDSBF	CNN	DS2RNN	SmDFCNN
500	8.3	9.3	8.7	7.2	6.5
1000	8.8	9.6	9.2	8.7	7.2
1500	10.3	9.8	9.1	8.7	7.6
2000	11.2	11.1	10.3	9.8	8.1
2500	11.8	11.4	10.2	9.5	8.5

Table VI mates timing issues arising from multiple methods and predicts method timing issues with the fewest. Lag was determined as the total time to load the database and perform feature selection and Classification within the specified time. Calculate the time complexity in milliseconds. The front page progressive code O(n) is the lower bound when executing instructions. Using time calculations and the worst form, the difference between the mean upper bound g(n) and the mean median finite time f(n) to calculate the mean time is a measure of how long it takes.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6s
DOI: https://doi.org/10.17762/ijritcc.v11i6s.6957
***Article Received**: 26 March 2023 **Revised**: 18 May 2023 **Accepted**: 01 June 2023*

V. CONCLUSION

To conclude, the evaluation of malicious web activities is analyzed to find the malicious activities through Selective Hyper Feature Link stability based on Soft-max Deep featured convolution neural network. This resultant proves the feature selection and Classification by analyzing the feature dependencies to reduce the dimension of the dataset. The URL specification features are also analyzed on post-link representation attributes to get hyperlink verification featured weights to classify the malicious sites. This proposed SmDFCNN system produces high performance in classification accuracy, produces 96.1 %, sensitivity produces 93.9 %, and specificity at 94.8%, higher than the previous system in malicious website detection by attaining high accuracy. This identifies the behavioral activity based on the performance features by getting parameter access in a non-residual state of response doing malfunctions. The result shows the comparison of performance in precision, recall rate, and detection accuracy as well as better performance than other methods.

REFERENCES

[1] J. Clayton and K. Bishal, “Towards detecting and classifying malicious URL using deep learning," Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), vol. 11, no. 4, pp. 31–48, 2020

[2] H. Lin, S. Cao, J. Wu, Z. Cao, and F. Wang, "Identifying Application-Layer DDoS Attacks Based on Request Rhythm Matrices," in IEEE Access, vol. 7, pp. 164480-164491, 2019, doi: 10.1109/ACCESS.2019.2950820.

[3] R. B. Basnet and R. Shash, “Towards detecting and classifying network intrusion traffic using deep learning frameworks,” Journal of Internet Services and Information Security, vol. 9, no. 4, pp. 1–17, 2019.

[4] Y. Qin, W. Wang, S. Zhang and K. Chen, "An Exploit Kits Detection Approach Based on HTTP Message Graph," in IEEE Transactions on Information Forensics and Security, vol. 16, pp. 3387-3400, 2021, doi: 10.1109/TIFS.2021.3080082.

[5] T. Li, C. Ren, Y. Fu, J. Xu, J. Guo and X. Chen, "Webshell Detection Based on the Word Attention Mechanism," in IEEE Access, vol. 7, pp. 185140-185147, 2019, doi: 10.1109/ACCESS.2019.2959950.

[6] Z. Feng, C. Shuo, and W. Xiaochuan, “Classification for DGA-based malicious domain names with deep learning architectures,” in Proceedings of the Second International Conference on Applied Mathematics and information technology, p. 5, Shanghai, China, October 2017.

[7] X. Liu and J. Fu, "SPWalk: Similar Property Oriented Feature Learning for Phishing Detection," in IEEE Access, vol. 8, pp. 87031-87045, 2020, doi: 10.1109/ACCESS.2020.2992381.

[8] C. Choudhary, R. Sivaguru, M. Pereira, B. Yu, A. Nascimento, and M. De Cock, “Algorithmically generated domain detection and malware family classification,” in Proceedings of the International Symposium on Security in Computing and Communication, pp. 640–655, Springer, Singapore, September 2018.

[9] I. Vaccari, S. Narteni, M. Aiello, M. Mongelli and E. Cambiaso, "Exploiting Internet of Things Protocols for Malicious Data Exfiltration Activities," in IEEE Access, vol. 9, pp. 104261-104280, 2021, doi: 10.1109/ACCESS.2021.3099642.

[10] X. Ruan, Z. Wu, H. Wang and S. Jajodia, "Profiling Online Social Behaviors for Compromised Account Detection," in IEEE Transactions on Information Forensics and Security, vol. 11, no. 1, pp. 176-187, Jan. 2016, doi: 10.1109/TIFS.2015.2482465.

[11] C. Amrutkar, Y. S. Kim and P. Traynor, "Detecting Mobile Malicious Webpages in Real Time," in IEEE Transactions on Mobile Computing, vol. 16, no. 8, pp. 2184-2197, 1 Aug. 2017, doi: 10.1109/TMC.2016.2575828.

[12] T. Shibahara, K. Yamanishi, and Y. Takata, “Malicious URL sequence detection using event de-noising convolutional neural network,” in Proceeings of the Communications (ICC), 2017 IEEE International Conference, pp. 1–7, IEEE, Paris, France, May 2017.

[13] J. C. Prieto, A. Fernández-Isabel, I. M. De Diego, F. Ortega and J. M. Moguerza, "Knowledge-Based Approach to Detect Potentially Risky Websites," in IEEE Access, vol. 9, pp. 11633-11643, 2021, doi: 10.1109/ACCESS.2021.3051374.

[14] D. R. Patil and J. B. Patil, “Survey on malicious web pages detection techniques,” International Journal of U- and E-Service, Science and Technology, vol. 8, no. 5, pp. 195–206, 2015.

[15] C. Xu, J. Shen, and X. Du, “Detection method of domain names generated by DGAs based on semantic representation and deep neural network,” Computers & Security, vol. 85, pp. 77–88, 2019.

[16] R. -H. Hwang, M. -C. Peng, C. -W. Huang, P. -C. Lin and V. -L. Nguyen, "An Unsupervised Deep Learning Model for Early Network Traffic Anomaly Detection," in IEEE Access, vol. 8, pp. 30387-30399, 2020, doi: 10.1109/ACCESS.2020.2973023.

[17] Srinivasan, Sriram& Ravi, Vinayakumar&Arunachalam, Ajay &Alazab, Mamoun&Kp, Soman. (2021). DURLD: Malicious URL Detection Using Deep Learning-Based Character Level Representations. 10.1007/978-3-030-62582-5_21.

[18] D. Lazar, K. Cohen, A. Freund, A. Bartik and A. Ron, "IMDoC: Identification of Malicious Domain Campaigns via DNS and Communicating Files," in IEEE Access, vol. 9, pp. 45242-45258, 2021, doi: 10.1109/ACCESS.2021.3066957.

[19] X. Yan, Y. Xu, B. Cui, S. Zhang, T. Guo and C. Li, "Learning URL Embedding for Malicious Website Detection," in IEEE Transactions on Industrial

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6s
DOI: https://doi.org/10.17762/ijritcc.v11i6s.6957
***Article Received**: 26 March 2023 **Revised**: 18 May 2023 **Accepted**: 01 June 2023*

Informatics, vol. 16, no. 10, pp. 6673-6681, Oct. 2020, doi: 10.1109/TII.2020.2977886.

[20] J. Yang, P. Yang, X. Jin, and Q. Ma, "Multi-classification for malicious URL based on improved semi-supervised algorithm," in Proceedings of the IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), pp. 143–150, IEEE, Guangzhou, China, July 2017.

[21] D. Liu and J. -H. Lee, "CNN Based Malicious Website Detection by Invalidating Multiple Web Spams," in IEEE Access, vol. 8, pp. 97258-97266, 2020, doi: 10.1109/ACCESS.2020.2995157.

[22] G. K., S. ., Reddy K. R., U. K. ., A. Rathod, J. ., B. P., S. ., & P. K., L. . (2023). Diabetic Retinopathy Prediction using Modified Inception V3 Model Structure. International Journal of Intelligent Systems and Applications in Engineering, 11(1), 261–268. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2466