Phishing URL Classification Analysis Using ANN Algorithm

Conference Paper · September 2021

DOI: 10.1109/GUCON50781.2021.9573797

4 authors, including:

Krishna Mridha
Case Western Reserve University School of Medicine
40 PUBLICATIONS **482** CITATIONS
SEE PROFILE

Dinesh Saravanan
SNS College of Technology
15 PUBLICATIONS **126** CITATIONS
SEE PROFILE

Ankush Ghosh
Chandigarh University
148 PUBLICATIONS **2,655** CITATIONS
SEE PROFILE

Phishing URL Classification Analysis Using ANN Algorithm

Krishna Mridha
Computer Engineering
Marwadi University
Rajkot, Gujarat, India
krishna.mridha@marwadiuniversit.ac.in

Jahid Hasan
Computer Science
Iowa State University
jhasan@iastate.edu

Rabindra Nath Shaw
School of Electrical, Electronics, and Communicaton Engineering
Galgotias University
r.n.s@ieee.org

***Abstract*—Over the years in social engineering, phishing URLs became a significant threat for internet users as most cyber-crime or attacks redirect to individual target users by sending a malicious or crafted URL which allows the attacker to capture sensitive pieces of information of the victim users. According to this research, most of the existing phishing detection tools showed an overall accuracy level of 70% to 92.52%. This paper introduced a random forest (RF) and artificial neural network(ANN) based machine learning (ML) algorithmic classification models for detecting phishing URLs effectively with higher accuracy. The experimental results showed that the proposed RF and ANN models could classify phishing URLs' legitimacy labels with an efficiency rate of up to 99%, much more significant than traditional ways. In this paper, this comparison's algorithmic learning models demonstrate that the proposed ANN classification holds a competent accuracy label of 98.72%.**

Keywords— Phishing, URLs, Random Forest, ANN, Machine Learning.

I. Introduction

Phishing is a form of cyber-crime attack that enormously causes personal or organizational financial losses at higher risk by the attacker sending crafted fake URLs to access a website that seems to be legitimate, but in reality, it is not. It leaks sensitive credentials like email accounts, bank passwords, cell numbers, bank card details, social security numbers (SSN), and much more other valuable data [8-12]. Recent COVID-19 outbreaks resulted in pandemic isolation, where most people staying at home and browsing the internet each day increased users' online activity. This continuous use of the internet may occur data breaches of internet users. The perpetrator can find it a lot easier to send "Phishing URLs" to the victims through social networking sites, emails, online advertisements, or any other forms.

There some other research is ongoing to avoid such situations, including the emergence of machine learning approaches. There is numerous software used to detect phishing URLs. However, it is not well-defined that how reliable those tools are to recognize legitimate/phishing URLs. Here in this paper, we have demonstrated a unique model that can efficiently predict the phishing URL. We have used random forest (RF), artificial neural network (ANN) based machine learning (ML), and deep learning (DL) algorithms [13] to develop the resolution. Applying such algorithms involves comparing their accuracy to other methods and demonstrating how to enhance efficiency by using a deep learning method. The other researchers are investigating further into this area; though, the problem was to achieve the desired precision. Our experimental findings from the datasets were efficient to attain the highest level of accuracy.

The rest of the paper is structured as follows: Section II presented a similar study and its implications. Section III provides more details of the methodology of machine learning algorithms and their approaches. Analyze the datasets and include a performance assessment of the results in section IV. Finally, section V summarizes the entire paper.

II. Related Work

M. Aburrous et al. [1] developed a data mining system for e-banking phishing: mobile malicious software studies and malicious software experiments to evaluate user perception. These investigations have shown a variety of phishing traits and markers. They studied six different classifiers and identified that the Multi-Class Classification based on Association Rule (MCCAR) technique obtained the best completeness.

I. Rahmi et al. [2] proposed a hybrid feature selection that optimized phishing email detection; the researchers used dual feature extraction and implementation methods. Feature extraction is used to eliminate inappropriate features that could adversely impact classification results. Features can be extracted from the computerized document and body, followed by feature selection using the Hybrid Feature Selection (HSF) method, and subsequently, classification adopting the Bayes Net algorithm.

Andronicus A et al. [3] investigate and analyzed 15 bogus websites from different studies and applied a feature selection method converged on the essential Information Gain (IG) of certain classified traits, with 14 of them was chosen. For the mail spam detection method, the Random Forest (RF) classification was used; and they found that their research achieved the most significant precision over bulky datasets.

R.Sumathi et al. [4] working optimization system was used as a classification model. Researchers have noticed that this classification had the highest performance in precision and reliability than other standard data mining algorithms.

J. Shad et al. [5] developed a machine-learning algorithm to detect malicious websites. They also calculated the efficacy of five machine learning techniques: Decision Tree (DT), Random Forest (RF), Gradient Boosting, Generalized Linear Model (GLM), and Generalized Additive Model (GAM) (GAM). The classifications of each algorithm's accuracy, recall, and precision was calculated and compared. The random forest approach achieved the highest degree of accuracy (98.4 percent), recall (98.59 percent), and precision (97.70 percent).

Considering most recent phishing attacks use broadcast email (spam) to pull victims into visiting a phishing website, the initial step is to dodge phishing at the email level. Another choice is to use a security extension toolbar. A phishing filter toolbar solution with extended features, such as blocking the user's interaction with a classified phishing website, is available in Internet Explorer 8. The objective of this research is to determine whether a web link URL is legitimate or phishing. If we can efficiently recognize phishing URLs, we can secure our confidential data from being leaked.

III. Methodology

A function selection algorithm is used in the proposed machine learning-based model (fig. 1) to improve accuracy. The machine learning repository has provided 30 attributes with a series of phishing websites databases. The algorithm selects certain features from the initial dataset that are crucial in shaping the prediction's outcome by filtering through 30 features [14]. As a result, by including a few features, unrelated features have little impact on the model's accuracy or prediction. Moreover, the prediction model is trained using ensemble learning, which employs several learning models. When making predictions, using different models ensures that a single model does not bias the results. For instance, all of the models perform that one particular URL is phishing, then definitely the prediction will be phishing.

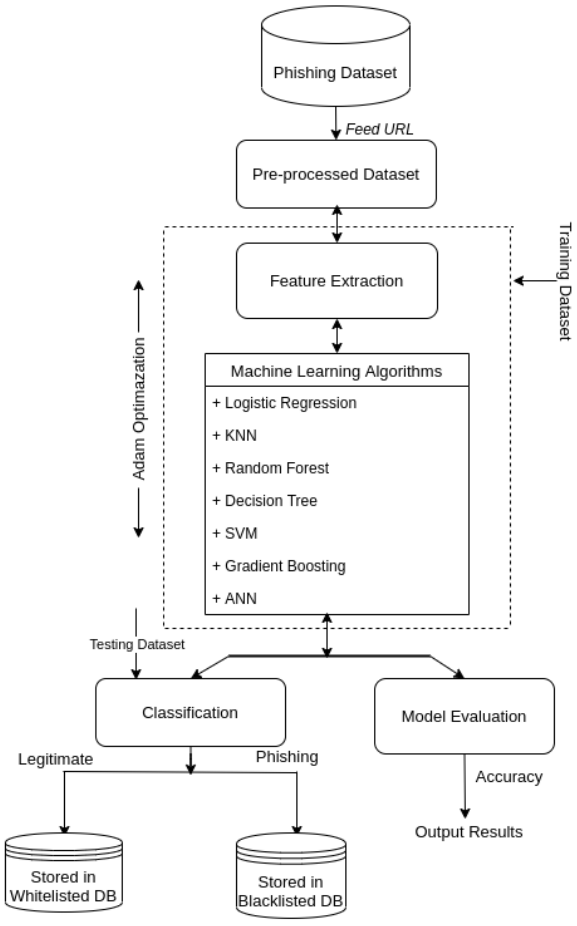

Figure 1: Proposed Model Architecture

A. *Uniform Resource Locator (URL) And Its Features*

It mainly refers to a "unique address of the website," published or located on the web. This resource contains HTML pages, CSS scripts, and images. Below fig. 2 displays the structure of a URL:

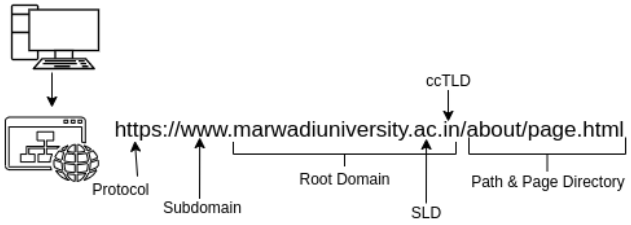

Figure 2: Domain Analysis

The sub-domain portions are entirely under the attacker's control, who can set any value they want [15-17]. The URL may also contain route and file items that the phisher may change at any time. We will refer to certain areas of the URL as FreeURL in the rest of the body. The attacker has the authority to register any previous non-registered web domain. The phisher can modify FreeURL at any stage to generate a unique URL. Security defenders have difficulty identifying phishing domains depending on the domain (the FreeURL).

There are 4 types of features that we can extract from URLs.

i. Address Bar Features.
ii. Abnormal Features.
I. HTML and JavaScript Features.
II. Domain Features.

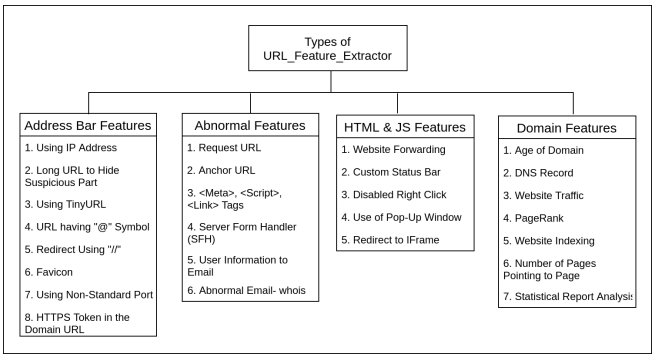

Figure 3: URL_Feature_Extraction_Diagram

B. *Data Preprocessing*

Noise, missing values, and uncertainty are all typical issues with raw data. The data consistency affects the data mining outcomes. To assist in the enhancement of data quality and, as a result, the mining process. Data pre-processing is essential in the data mining process since it deals with the initial datasets planning and transformation:

a) In the data cleaning section, we check our raw dataset for missing values, and it has no missing values in the dataset.
b) We drop our extra column, namely "*index*" from 32 features. From the remaining 31 features, the independent (x) features are 30, and the rest are dependent (y) features.

c) We mapped our two classes from (*-1 to 0*) and from (*1 to 1*).

Here, class *0* for "*Legitimate*" and *1* for "*Phishing*". The final process is the data transformation process that used a normalized function. Data transformation is the process of transforming or combining data into formats that are suitable for mining. The attribute data is weighted to fit into a narrow range, including the (*-1.0 to 1.0*) or (*0 to 1.0*).

C. *Trian and Test Split*

The train-test split method predicts machine learning techniques as they're used to make decisions and predictions to train the model. The total number of items in the original dataset is *11055*, and the number of features is *31*. As a whole, the scale of the feature matrix is *11055x30* (the rest of one feature is for classification level). We split our entire dataset into two subsets based on it, such as the train and test sets

D. *Learning Algorithms*

The following task is to develop a classification model based on the machine learning approach to classify two sets of labels. Usually, all models depend on a feature matrix generated from the data acquired from the acquisition system. Since no thumb rule determines an algorithm for the available data to be classified, several algorithms experimented with the dataset. In this research, we will refer to the following two classifiers:

1. ***Random Forest:*** The random forest is a classified an algorithm that uses several decision trees to classify data. When constructing each tree, it employs bagging and attributes random nature to establish a negatively correlated forest of trees whose working group estimation is more reliable than that of any single tree.

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)} \quad (1)$$

Where,

ni_j is the importance of node j

w_j is a weighted number of samples reaching node j

C_j is the impurity value of Node j

$left(j)$ is child node form node split on node j

$right(j)$ is child node from node split on node j

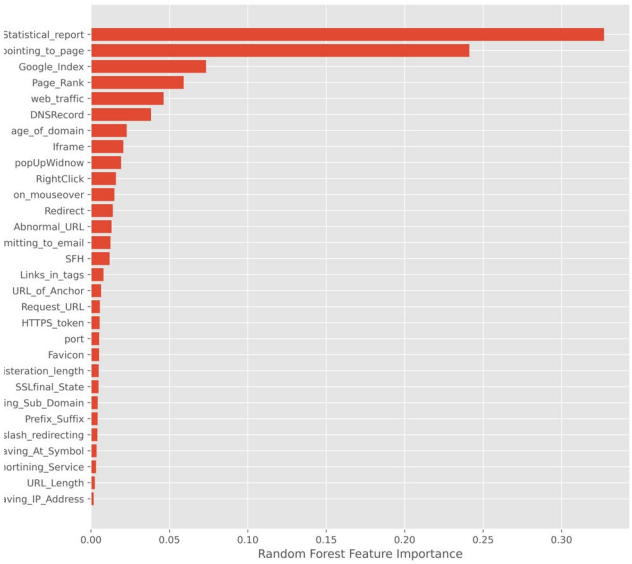

Figure 4: Features Importance by Random Forest Classification Model

From the Random Forest, we can predict our class but here we also get the probability of this particular class. Bypassing the parameter "type = prob" to the Random Forest set, we get the probability rather than the expected class of the data point. This is the probability first theorem define as Chebyshev inequality. It X is a random variable with standard deviation σ and means μ, then for any $\in$>0 [12].

$$P(|X-\mu|>\in) <= \frac{\sigma^2}{\in^2} \quad (2)$$

In theorem number two name as Bounded convergence. This theorem is like that we are given a sequence $h_1(x), h_{2(x)}$ …of function. With $h_k(x) \le M$ for fixed $M > 0$ defined on a space S of finite measure then [12],

$$\lim_{x\to\infty} \int_S dx h_k(x) \underset{k\to\infty}{\longrightarrow} \int_S dx \lim_{k\to\infty} h_k(x) \quad (3)$$

2. *Artificial Neural Network*

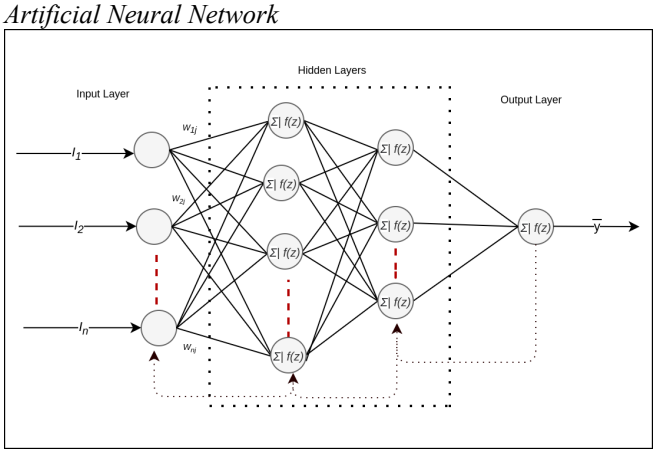

Figure 5: ANN Model Design

The smallest neural network is the perceptron, which has n parameters, an only neuron, but only one outcome, where n represents the number of attributes in our dataset. Forward propagation is the method of transmitting data across a neural network, and it is done in a perceptron.

Multiplication the inputs value with the weights and add the results. For instance, if our row vectors of the features and weights are $x = [x_1, x_2, ..., x_n]$ and $w = [w_1, w_2, ..., w_n]$. Then their dot product should be like this.

$$\sum = x.w = (x_1 w_1 + x_2 w_2 + ... + x_n w_n) \quad (1)$$

For moving left to right, we have to add bias b to the desired result. So the equation should be like this. These values will be store in one variable called z for further use.

$$z = x.w + b \quad (2)$$

Now, we have to pass the value z to a non-linear activation function. Even though, we have so many activation functions for a different problem. In this article, the sigmoid activation function is being used because we are going to predict the probability of two different classes [ ]. The forecast is contrasted to the true value to determine its accuracy. The real worth is a 1 if the data belongs to that class, otherwise, it is a 0.

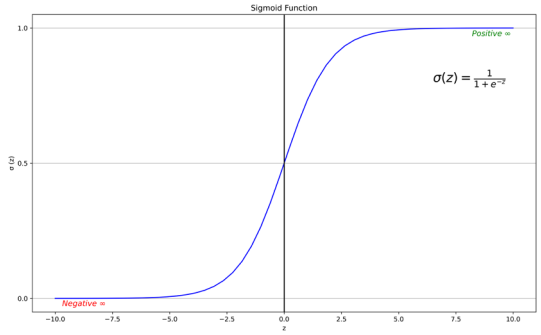

Figure 6: Sigmoid Activation Function Diagram

$$\overline{Y} = \sigma(z) = \frac{1}{1+e^{-z}} \quad (3)$$

Where σ is the sigmoid activation and after forwarding propagation whatever result we get is known as predictable result denote as $\overline{Y}$? Similar calculations are done at each hidden layer and the values are passed into the output layer.

Learning Algorithm is the next term which is consists of two parts namely Backpropagation and Optimization. In the following two steps, we'll go over how a perceptron performs backpropagation.

The first step, A loss function is used to determine how far we are from our desired solution. Loss function Mean square error is the difference between actual y_i and predicted values $\overline{Y}_i$.

$$MSE_i = (y_i - \overline{Y})^2 \quad (4)$$

The loss function is calculated for the entire training dataset and their average is called the Cost function C.

$$C = MSE_i = \frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{Y})^2 \quad (5)$$

We also need to understand how well the cost function varies in respect to weights and bias to find the perfect weights and bias for our Perceptron. Gradients (rate of change) how one quantity varies in comparison to another are used to do this. In our case, we need to determine the cost function's gradient about the weights and bias.

$$\frac{\delta C}{\delta w_i} = \frac{\delta C}{\delta \overline{Y}} . \frac{\delta \overline{Y}}{\delta z} . \frac{\delta z}{\delta w_i} \quad (6)$$

Here, The gradient of the cost function (C) concerning the predicted value $\overline{Y}$ chain rule is like this.

$$\frac{\delta C}{\delta \overline{Y}} = \frac{\delta}{\delta x} . \frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{Y})^2$$

$$= 2\frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{Y})$$

$$= \frac{2}{n} sum(y_i - \overline{Y}) \quad (7)$$

Now, we have to find out the gradient of the predicted value concerning z .

$$\frac{\delta \overline{Y}}{\delta z} = \frac{\delta}{\delta z}\sigma(z)$$

$$= \sigma(z)(1 - \sigma(z)) \quad (8)$$

Here, the gradient of z with the respect of w_i .

$$\frac{\delta z}{\delta w_i} = \frac{\delta}{\delta w_i}(z)$$

$$= \frac{\delta}{\delta w_i}\sum_{i=1}^{n}(x_i . w_i) \ = x_i \quad (9)$$

Finally, we get,

$$\frac{\delta c}{\delta w_i} = \frac{2}{n} sum((y_i - \overline{Y})(\sigma(z)(1 - \sigma(z)))(x_i)) \quad (10)$$

Second is the optimization. The best factor from a set of possible options is selected, in this scenario, the appropriate weights, and bias of the neuron. In this article, we are using adaptive moment estimation (adam). Adam is nothing but the combination of other two optimizations namely Momentum and RMSprop.

$$w_{t+1} = w_t - \frac{\infty}{\sqrt{\hat{s}_t + \in}} . \hat{v}_t \quad (11)$$

Where,

$$\hat{v}_t = \frac{v_t}{1-\beta_1^t}$$

$$\hat{s}_t = \frac{v_t}{1-\beta_2^t}$$

They are called Bias Correction

Again, for the reference Bias Correction equation we know,

$$v_t = \beta_1 v_{t-1} + (1-\beta_1)\frac{\delta l}{\delta w_t} \quad (12)$$

$$s_t = \beta_2 s_{t-1} + (1-\beta_2)[\frac{\delta l}{\delta w_t}]^2 \quad (13)$$

IV. DATASETS & PERFORMANCE ANALYSIS

Using supervised learning techniques, the malware websites forecasting approach is formulated. The research dataset was obtained from *KAGGLE* [7]. The attributes that define the characteristics of websites are derived in section III. The class label (*-1*) is allocated to the input vectors referring to the phishing website, while (*+1*) is assigned to the legitimate website.

The appropriate training as mentioned above set is used to build machine learning models. 10-fold cross-validation is adopted to adjust the tuning of the algorithms' hyper-parameters and the model's robustness. The average accuracy of a 10-fold training and testing set and the standard deviation in the 10-fold system of selecting training and testing sets shows in Table 1.

Table 1: Comparison between k-fold accuracy vs without cross-validation accuracy

Machine learning algorithms	K-fold Mean Ac (%)	Accuracy (%)	Std (%)
Gradient Booster	94.53	95.67	0.46
Support Vector Machine	94.52	95.61	0.32
Logistic Regression	92.30	96.05	0.83
K-Nearest Neighbors	94.00	95.73	0.36
Decision Tree	90.02	96.61	0.70
Random Forest	97.04	97.65	0.55
Artificial Neural Network	**98.45**	**98.72**	0.62

Table 1 shows that the Random Forest algorithm not only performs best with this data 1-fold but also experiences the least variance when conducting k-fold testing and test data collection, as well as subsequent training. On the other hand, the Artificial Neural Intelligence Algorithm is being used for better performance. From Table 1, it is concluded that among all of those algorithms ANN performs the best accuracy not even only accuracy but also cross-validation.

Table 2. Performance measures like Precision, Recall, and F1-score.

	Classes	Gradient Booster	KNN	Logistic	Decision Tree	Random F.	SVC	Ann
Precision	0	0.94	0.94	0.92	0.96	0.97	0.94	0.97
	1	0.97	0.97	0.95	0.97	0.98	0.97	1.00
Average:		0.96	0.96	0.94	0.97	**0.98**	0.96	**0.99**
	Classes	Gradient Booster	KNN	Logistic	Decision Tree	Random F.	SVC	Ann
Recall	0	0.96	0.96	0.94	0.96	0.98	0.96	0.99
	1	0.96	0.96	0.94	0.97	0.97	0.85	1.00
		0.96	0.96	0.94	0.96	**0.98**	0.90	**1.0**
	Classes	Gradient Booster	KNN	Logistic	Decision Tree	Random F.	SVC	Ann
F1-score	0	0.95	0.95	0.93	0.96	0.97	0.95	1.00
	1	0.96	0.96	0.94	0.97	0.98	0.96	0.96
Average:		0.96	0.96	0.93	0.97	**0.98**	0.96	**0.98**

Table 2 shows other performance measures of a classifier like precision, recall, and F1 scores for all different algorithms applied to both classes. The ANN method shows superiority by yielding 100% recall, precision. In other cases of Random Forest, it also shows the best recall, precision, and, F1-score among all of the Machine Learning algorithms. This suggests that the possibility of type-1 error (i.e., false positive, that shows class2 is wrongly detected as class1), and type-2 error (i.e., false negative, that shows class1 is wrongly detected as class1) are almost NIL. Subsequently, the F1-score which is computed by taking the harmonic mean of the precision and recall is 1.00.

Table 3: Errors Counts

Eros	Classifications						
	Gradient Booster	KNN	Logistic	Decision Tree	Random F.	SVC	Ann
MSE	0.044	0.042	0.062	0.033	**0.023**	0.043	**0.019**
MAE	0.044	0.042	0.062	0.033	**0.023**	0.043	**0.019**
RMS	0.210	0.207	0.250	0.184	**0.153**	0.201	**0.130**
MAPE	0.565	0.567	0.563	0.562	**0.565**	0.572	**0.407**

According to the table, the ANN and Random Forest have very low errors. In this section, we compute Mean Square Error (MSE), Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2 \tag{13}$$

$$MAE = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} = \frac{\sum_{i=1}^{n}|e_i|}{n} \tag{14}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{y_i - \hat{y}}{y_i} \tag{16}$$

Where,

y_i are the actual values

$\hat{y}$ are the predicted values

n is the number of non-missing data points

i is the variable for iteration

Figure 7 showing the loss between training data and validation data. The x-axis measures the number of epoch and the y-axis measure the loss values. We conclude that the number of the epoch is increasing the loss is decrease. We have taken 136 epochs and after that loss is being increased. The loss between train and validation data is almost the same.

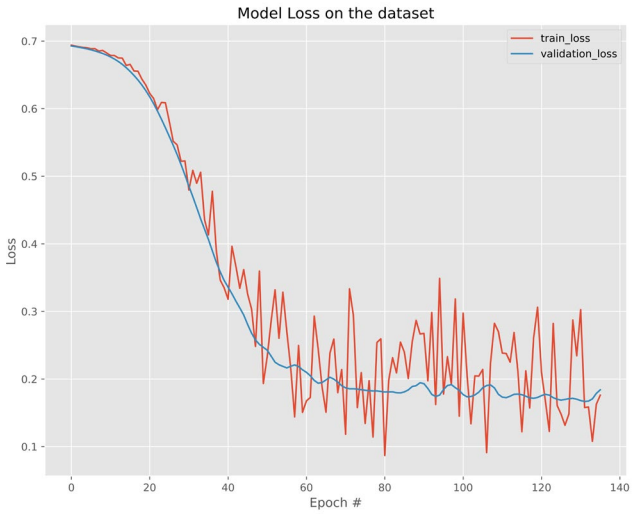

Figure 7: Train vs Validation loss

Figure 8 showing the accuracy between training data and validation data. The x-axis measures the number of epoch and the y-axis measure the accuracy values. We conclude that the number of epochs is increasing the accuracy is decreased. The accuracy between train and validation data is almost the same.

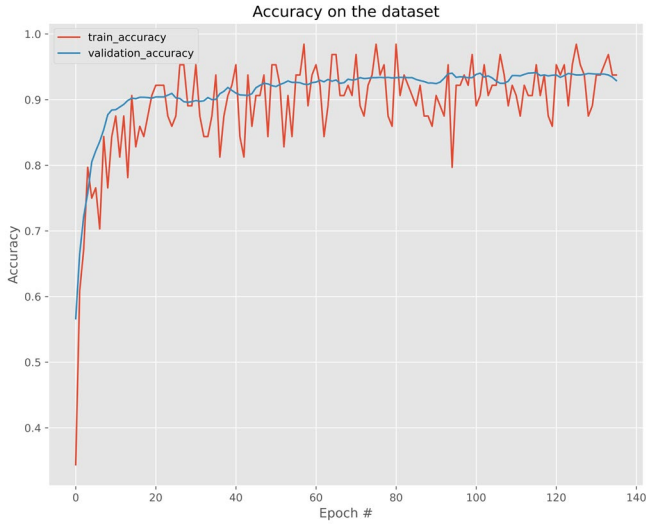

Figure 8: Train vs Validation Accuracy

V. Conclusion

Conclusively, this research proposed phishing website detection as a classification algorithm and demonstrates how a machine learning approach can also be used to forecast whether a crafted URL is legitimate or phishing. Both the RF and the ANN models have been used to train the prediction models. The performance of the proposed model was estimated using 10-fold cross-validation and then analyzed without cross-validation. According to both the findings, the RF and ANN classifiers perform better than some other five models. Additionally, in a later project, we will develop an NLP and GUI-based web browser extension framework to gain significantly greater precision to lessen the phishing problems in real-time applications.

References

[1] M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah, "Predicting Phishing Websites using Classification Mining Techniques with Experimental Case Studies," 2010 Seventh International Conference on Information Technology: New Generations, Las Vegas, NV, USA, 2010, pp. 176-181. DOI: https://doi.org/10.1109/ITNG.2010.117.

[2] Isredza Rahmi, A Hamid, Jemal Abawajy, Tai-hoon Kim, "Using feature selection and classification scheme for automating phishing email detection," Studies in informatics and control, ISSN 1220-1766, (1), pp. 61-70, 2013. DOI: https://doi.org/10.24846/v22i2y10307.

[3] Andronicus A. Akinyelu and Aderemi O. Adewumi., "Classification of Phishing Email Using Random Forest Machine Learning Technique," Hindawi Publishing Corporation, Journal of Applied (SI16) 1-6, 2014. DOI: https://doi.org/10.1155/2014/425731.

[4] R.Sumathi and Mr.R.Vidhya Prakash, "Prediction of Phishing Websites Using Optimization Techniques," International Journal of Moder Engineering Research (IJMER), 2012.

[5] I. Tyagi, J. Shad, S. Sharma, S. Gaur, and G. Kaur, "A Novel Machine Learning Approach to Detect Phishing Websites," 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2018, pp. 425–430. DOI: https://doi.org/10.1109/SPIN.2018.8474040.

[6] Rana Singh. (Oct 9, 2019). Mathematics behind Random forest and XGBoost. Available Online: https://medium.com/analytics-vidhya/mathematics-behind-random-forest-and-xgboost-ea8596657275

[7] Akash Kumar, Phishing website dataset, Kaggle. Available online: https://www.kaggle.com/akashkr/phishing-website-dataset.

[8] Xi Zhang, Yu Zeng, Xiao-Bo Jin, Zhi-Wei Yan, Guang-Gang Geng, "Boosting the phishing detection performance by semantic analysis", Big Data (Big Data) 2017 IEEE International Conference on, pp. 1063-1070, 2017.

[9] Ying Xue, Yang Li, Yuangang Yao, Xianghui Zhao, Jianyi Liu, Ru Zhang, "Phishing sites detection based on Url Correlation", Cloud Computing and Intelligence Systems (CCIS) 2016 4th International Conference on, pp. 244-248, 2016.

[10] Swati Gupta, Sagun Sodhani, Dhaval Patel, Biplab Banerjee, "News Category Network-based Approach for News Source Recommendations", Advances in Computing Communications and Informatics (ICACCI) 2018 International Conference on, pp. 133-138, 2018.

[11] Gilchan Park, Julia Rayz, "Ontological Detection of Phishing Emails", Systems Man and Cybernetics (SMC) 2018 IEEE International Conference on, pp. 2858-2863, 2018.

[12] AlMaha Abu Zuraiq, Mohammed Alkasassbeh, "Review: Phishing Detection Approaches", Trends in Computing Sciences (ICTCS) 2019 2nd International Conference on new, pp. 1-6, 2019.

[13] Avisha Das, Shahryar Baki, Ayman El Aassal, Rakesh Verma, Arthur Dunbar, "SoK: A Comprehensive Reexamination of Phishing Research From the Security Perspective", Communications Surveys & Tutorials IEEE, vol. 22, no. 1, pp. 671-708, 2020.

[14] Jitendra Kumar, A. Santhanavijayan, B. Janet, Balaji Rajendran, B.S. Bindhumadhava, "Phishing Website Classification and Detection Using Machine Learning", Computer Communication and Informatics (ICCCI) 2020 International Conference on, pp. 1-6, 2020.

[15] Hamza H. M. Altarturi, Nor Badrul Anuar, "A preliminary study of cyber parental control and its methods", Application Information and Network Security (AINS) 2020 IEEE Conference on, pp. 53-57, 2020.

[16] K. Archana Janani, V. Vetriselvi, Ranjani Parthasarathi, G. Subrahmanya VRK Rao, International Conference on Computer Networks and Communication Technologies, vol. 15, pp. 217, 2019.

[17] Kalyan Nagaraj, Biplab Bhattacharjee, Amulyashree Sridhar, Sharvani GS, "Detection of phishing websites using a novel twofold ensemble model", Journal of Systems and Information Technology, 2018.