Some studies only consider the URL, while some only consider the Email header and body. Here, we will consider both the email text and the URL in the email.

RNN, CNN and SVM will be used to train and test the data and the results between them will be compared.

The dataset will be explained first after which the three algorithms will be described.

1. Dataset:

Most studies consider only either the email content, i.e., the email header or email body, while other studies consider only the URL. In this study we will use both the email content as well as the URLs.

The datasets we will be using for this study are

1. SpamAssassin (<https://spamassassin.apache.org/old/publiccorpus/obsolete/>), a collection of spam and ham emails. Ham emails are the emails that are not spam.
2. Phishcorpus (<https://monkey.org/~jose/phishing/>), a collection of phishing emails
3. Website phishing dataset (<https://archive.ics.uci.edu/ml/datasets/Website+Phishing>), a collection of phishing and legitimate URLs.

From the emails datasets, we will be extracting 3000 emails of both phishing and ham category randomly. Spam emails are not considered in this study because they are usually advertisements, contain more content than a regular email, use unusual HTML markup and may use coloured text.

All three datasets are public datasets. SpamAssassin and Phishcorpus are a collection of emails over several years. SpamAssassin has emails collected from 2004-2005, while Phishcorpus has emails collected from 2015-2019.

The Website phishing dataset has 1353 instances, 805 of which are phishing URLs from Phishtank, a website where users can report and track suspicious phishing websites, 584 are legitimate websites from Yahoo. This data was collected by the creator using a web script in PHP (bibex)

These datasets are widely used in studies for phishing detection, contain sufficient information to difference between phishing and legitimate emails, hence they are appropriate to be used in this study.

TABLE WITH THE NUMBERS OF DATA USED IN EACH DATASET

Since SpamAssassin and Phishcorpus consist of only emails, each email will be parsed to extract features such as:

Email header

Sender Domain

Number of weblinks

If they contain javascript

Number of Spelling errors

Phishing indicator words

Phishing emails usually have spelling errors and improper grammatical structure, hence the words in the emails will be checked for any unusual spellings. They also contain certain words that causes a sense of urgency to the reader, for example, words like “Password expired”, “bank account details”, “respond in 24 hours”, etc. These kind of words will be added after finding the most common words from the phishing emails.

The Website phishing dataset consists of 10 features which are:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SFH** | popUpWidnow | SSLfinal\_State | Request\_URL | URL\_of\_Anchor | web\_traffic | URL\_Length | age\_of\_domain | having\_IP\_Address | Result |

(check paper and write the meaning)

DESCRIBE ALL FEATURES AND TYPES

Data pre-processing:

Email dataset:

1. Tokenization: In this step, we split all sentences in the email body to words.
2. Lower casing and Stop words removal: The tokenised words are all lower cased because “Email” and “email” are two different words according to a machine. All stop words are removed from the tokenised data because stop words like “a”, “the”, “of” do not convey any special meaning to our dataset.
3. Lemmatization: The word is transformed into the base form of the word. For example, “charging” is changed to “charge”. This helps reduce the word vector space.

We used “SpamAssassin” dataset [1] for benign emails

and “Phishcorpus” dataset [2] for phishing emails. The

SpamAssassin dataset is a public mail corpus which includes a

selection of mail messages

1. LSTM:

LSTM is a modified version of the Recurrent Neural Network to help solve the vanishing gradient problem of RNNs during backpropagation. the gradient is used to update the weights of a neural network and the equation for the same is given below:

(put the gradient update rule formula)

The vanishing gradient problem happens when the gradient becomes very small in some iterations. Since updated gradient is very small, the starting layers of the RNN does not learn and RNN does not remember long term sequences i.e., they have a short-term memory problem. LSTM is a solution to this problem. They have gates that help in regulation of information. The gates learn through various data passed into them and they can either keep or discard the data based on their importance. Hence, passing only data that is relevant.

The cell state and gates are integral parts of the model. The cell state can be considered the memory of the network. It carries relevant information during the sequence processing. As the model trains, information to the cell state is added or removed via the gates. Gates decide on the information that should be allowed on the cell state and they learn on what information is to be considered relevant or irrelevant. The gates contain sigmoid activation that brings the values between 0 and 1. If a number is multiplied by 0 it gets forgotten and if a number is multiplied by 1 it remains the same. Using this the network learns which data should be keep or forgotten.

The tanh activation function helps in regulation of the flow of values throughout the network and helps ensure the values are kept between -1 and 1 to avoid explosion of values

There are three different gates that help in regulation of flow of information in Long Short Term Memory Cell

called the forget gate, input gate and the output gate.

1. Forget gate: The forget gate decides which information should be remembered or forgotten. The previous hidden state and current input information is passed through the sigmoid function and if the value is closer to 0, the information is forgotten. If the value is closer to 1, the information is kept.
2. Input gate: The input gate is used to update the cell state. Similar to the forget gate, the previous hidden state and the current input information is passed through the sigmoid function and the value that is closer to 0 is considered not important, values closer to 1 is consider important. The previous hidden state and the current input information is passed through the tanh function as well to keep the values between -1 and 1 that helps in the regulation of the network. The outputs from the sigmoid and tanh functions are multiplied. The output from the sigmoid helps decide which output from the tanh function is relevant to keep.

Cell State: The information from the forget gate and input gate helps in the calculation of cell state. The cell state is pointwise multiplied with the forget vector. If the values are closer to 0 they are removed. This output is pointwise added to the input gate output and this output is value to which the cell state updates to, hence forming a new cell state.

1. Output gate: The output gate makes the decision of what the next hidden state should be. Hidden state has information of previous outputs and is used for predictions. The previous hidden state and current input information is passed through a sigmoid function and then the new cell state is passed through the tanh function. The outputs of the sigmoid and tanh function helps decide on which information the hidden state carries. This new output is the hidden state. This new hidden state and new cell state is carried over for the next update.

ADD FORMULAE FOR EACH!!!

Explain LSTM with figure. Give formula

1. CNN: A convolution is a mathematical operation of two functions $x()$ and $w()$ to production a third function $y()$. It helps in joining of two sets of information. A CNN has a convolution layer, pooling layer and fully connected layer. The convolution is used to extract features of the input data by applying a filter or kernel.Multiple feature maps are gotten by applying the filter. An activation function like Rectified Linear Unit (ReLU)

is applied to these feature maps to get a non-linear relation. ReLu computes $f(k) = max(0,k)$. The input data usually is added padding to make sure there is no shrinkage of the feature map and so that relevant information is not lost. A pooling layer is used between the convolutional layers. It helps in reduction of complexity in dimension and to keep relevant information from the convolutions. All nodes in one layer is fully connected to nodes in the other layers.

1. SVM: Support Vector Machine is a supervised machine learning algorithm. It is widely used for classification problems. In our case it helps us classify if an email is a phishing email or not. It creates a decision boundary called a hyperplane, that segregates a n-dimensional space where n is the number of features in our dataset, into classes. SVM selects vectors that are extreme which helps in the creation of the hyperplane. The extreme vectors are called support vectors. SVM can we used for both linearly separable data (data that is classified into two classes using one straight line) and non-linearly separable data (data that cannot be classified using a single straight line).

experiment design