Comparison of Deep and Traditional Learning Methods for Email Spam Filtering

Abdullah Sheneamer
Faculty of Computer Science & Information Technology
Jazan University
Jazan, Saudi Arabia

***Abstract*—Electronic mail, or email, is a method for communicating using the internet which is inexpensive, effective, and fast. Spam is a type of email where unwanted messages, usually unwanted commercial messages, are distributed in large quantities by a spammer. The objective of such behavior is to harm email users; these messages need to be detected and prevented from being sent to users in the first place. In order to filter these emails, the developers have used machine learning methods. This paper discusses different methods which are used deep learning methods such as a Convolutional Neural Network (*CNN*) and Long Short-Term Memory (*LSTM*) models with(out) a GloVe model in order to classify spam and non-spam messages. These models are only based on email data, and the extraction set of features is automatic. In addition, our work provides a comparison between traditional machine learning and deep learning algorithms on spam datasets to find out the best way to intrusion detection. The results indicate that deep learning offers improved performance of precision, recall, and accuracy. As far as we are aware, deep learning methods show great promise in being able to filter email spam, therefore we have performed a comparison of various deep learning methods with traditional machine learning methods. Using a benchmark dataset consisting of 5,243 spam and 16,872 not-spam and SMS messages, the highest achieved accuracy score is 96.52% using CNN with the GloVe model.**

***Keywords*—Spam filtering; machine learning; deep learning; LSTM; CNN**

I. Introduction

Email is an inexpensive, effective, and fast way to exchange messages using the Internet. Spam email is a type of email where unwanted messages [1], [2], [3], [4], usually related to unwanted commercial messages, are sent in huge quantities by a spammer. Spam email can contain malicious content, such as a phishing attack and/or malware. Despite considerable cybersecurity improvements and continuous development, spam email and malware damage caused by spam emails can prevent communication, create increased traffic, and waste users' time where the spam emails must be manually deleted. It is also possible to miss important email messages that are accidentally deleted when manually removing large numbers of spam messages.

Cybersecurity is now a hot topic in industrial information and operational technologies. The definition of cybersecurity is technologies and processes which are built to protect computer hardware, software, networks, and data from unauthorized access, vulnerabilities, terrorists, and hackers. Cybersecurity is the protection of the internet, information, and network-based digital equipment from unauthorized access and amendment [5]. For many years, machine learning classifiers have had a prominent role in intelligent system development.

Machine learning methods are more robust and have greater flexibility; consequently, there has been an expansion in security execution and improved defense systems from growing and advanced cyber threats. Machine learning is a technique used in different areas of information security. The aim of this paper is to develop the proof of concept for shallow machine and deep learning for spam datasets. We make a comparison between shallow machine and deep learning methods and the most accurate algorithm capable of distinguishing the spam emails which have the lowest error rate. To summarize, the main points this paper achieves are:

- We propose a CNN model with(out) a GloVe deep learning-based model framework to classify spam email.
- We propose an LSTM model with(out) a GloVe deep learning-based model framework.
- We provide a comparison between traditional machine learning and deep learning algorithms on spam datasets.

In this section, we introduce the various types of spam email. Related works are highlighted in Section 2. In Section 3, we discuss our methodology. In Section 4, we present an evaluation and comparison of our proposed method and report results. Finally, Section 5 presents our conclusion.

A. Types of Email Spams

We have defined spam as any unwanted message which may or not be malicious, that is, a scam or a fraud. Spams can be bulk messages, reaching millions of people daily.

- **Ads.** This is one of the most common types of spam, usually, several unwanted emails offering services or products are received.
- **Chain Letters.** Chain letters usually take the form of exciting stories such as "something bad will happen to you" and encourage the recipient to respond to the message so that the bad event will not occur.
- **Email Spoofing.** Spoof emails are related to phishing scams. This happens when the spammers or phishers attempt to trick the recipient by impersonating someone he/she knows.

- **Hoaxes.** The email spams offer and miracle promises, such as “get rich in less than a week". The spammer tries to direct the recipient’s email spam to a malicious website.
- **Money Scams.** The spammers send spam email promising easy money. This involves asking for money for poor families who have suffered losses as a result of a natural disaster.
- **Malware Warnings.** These types of email warn the recipient about a malware infection on his device, such as a virus. Spammers send an email which states that they have a solution to the problem and that the recipient must provide some information or download an attached file.
- **Porn Spam.** The spammer sends emails containing pornography. This is very common as the pornography market is very profitable. Spammers can create malicious emails using lustful images and videos.

II. Related Works

Drucker *et al.* [6] compared the support vector machine (SVM) algorithm with Ripper, Rocchio, and boosting decision tree algorithms to classify an email as spam, or not. They performed experiments on datasets and chose the best 1000 features; one dataset contained over 7000 features. SVM accuracy was good and required less training time. However, extra time was needed to search for the best features in the training algorithm.

Banday and Jan [7] studied the design of common statistical spam filters, including Naive Bayes, Term Frequency-Inverse Document Frequency, K-Nearest Neighbour, Support Vector Machine, and Bayes Additive Regression Tree. They performed experiments on e-mail datasets to evaluate each classifier for accuracy, recall, precision, etc. Additionally, they studied the effectiveness and limitations of various types of statistical filter to discern spam from legitimate emails.

Radhakrishnan and Vaidhehi [8] proposed email spam classification using the Naive Bayes and J48 Decision Tree which has a feature size of 400 attributes. Their results do not achieve maximum efficiency.

Suleiman and Naymat [9] used a method for detecting SMS and email spams using deep learning, Na Bayes, and Random Forest while they used the H2O platform in Weka. In addition, they showed that the Naive Bayes classifier has the best runtime but the is the in regard to performance. Random Forest is the best in precision, recall, F-measure, and accuracy.

Singh *et al.* [10] presented the solution and classification processes and combining classification technique of spam filtering to obtain improved spam filtering results. They used machine learning and engineering knowledge and applied the NB, KNN, SVM, Artificial Neural Network classification methods.

Kumar *et al.* [11] suggested a deep learning-based approach for detecting spam images using convolutional neural networks (CNN), which used a dataset with 810 natural images and 928 spam images.

Jain. *et al.* [12] suggested a system for detecting spam social media texts by using a hybrid technique; combining Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) network architectures. They use pre-trained embeddings. CNN extracts the n-gram features from the text, whereas LSTM detects the long-term dependencies. The model has better performance than the shallow neural network.

III. Methodology

A. Preprocssing

This is the first stage and processes incoming mail to the user in several sub-steps, as shown below.

1) Tokenization: This stage divides incoming emails into a sequence of representative meaningful words by removing punctuation, known as tokens.

2) Noise Removal: The data usually features an increased amount of noise and unwanted symbols and characters, e.g. stop words, numbers, alphanumeric words, white spaces, punctuation, etc. An example of stop words is “a", “an", “is" and “it". While the example of punctuation is “?", “!", “," and “;". The procedures usually are removed, converting all letters into lower/upper case and the removal of numbers, punctuation, white spaces, and stop words.

3) Stemming: Stemming is one method by which to normalize the word form. For example, go, went, going are considered the same word in the feature matrix. In addition, stemming removes suffixes from words such as (“ing", “ly", “es", “s", etc.).

4) Lemmatization: Lemmatization is a method by which to normalize the word form. This is similar to stemming, however, lemmatization converts the word to its root form and morphological analysis using vocabulary or a dictionary. For example, the lemma word “better" is “good".

B. Feature Extraction

The input must usually be integers or floats for machine and deep learning algorithms. Here some approaches are used to convert words to integers or floats.

1) Count Vectorizer: First, all email data is inputted into the Count vectorizer algorithm; the Count vectorizer keeps a dictionary of all words and their respective *ID*, which represents the count of the word.

2) TF-IDF Vectorizer: The TF-IDF Vectorizer is used to calculate word frequencies and is known as the TF-IDF. Term Frequency (TF): this summarises the frequency of a specific word appearing in a document. Inverse Document Frequency (IDF): this downscales words that frequently appear across documents. It tokenizes files, learns the vocabulary and inverse document frequency weightings, as well as encoding new files.

3) Word Embedding: Word embedding converts words to a vectorized format, which then represents the word’s position in a higher-dimensional space. The cosine distance of the two-word vectors is shorter and closer to each other if those two words have a similar meaning. For example, $King - Man + Woman = Queen$.

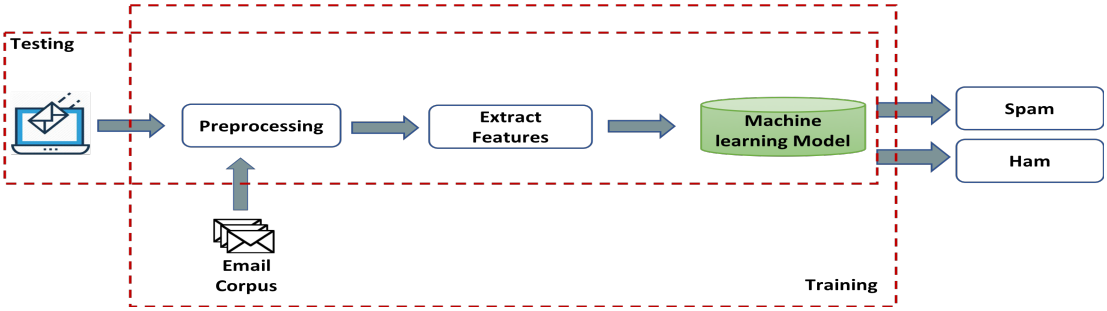


Fig. 1. General Traditional Machine Learning Model

TABLE I. BRIEF DESCRIPTION OF SOME ENRON EMAIL AND SMS SPAM COLLECTION DATASET

Dataset	# Total	# Ham (Not Spam)	# Spam
Enron1	5,172	3,672	1,500
Enron2	5,857	4,361	1,496
Enron3	5,512	4,012	1,500
SMSSpamCollection	5,574	4,827	747
All Emails	22,115	16,872	5,243

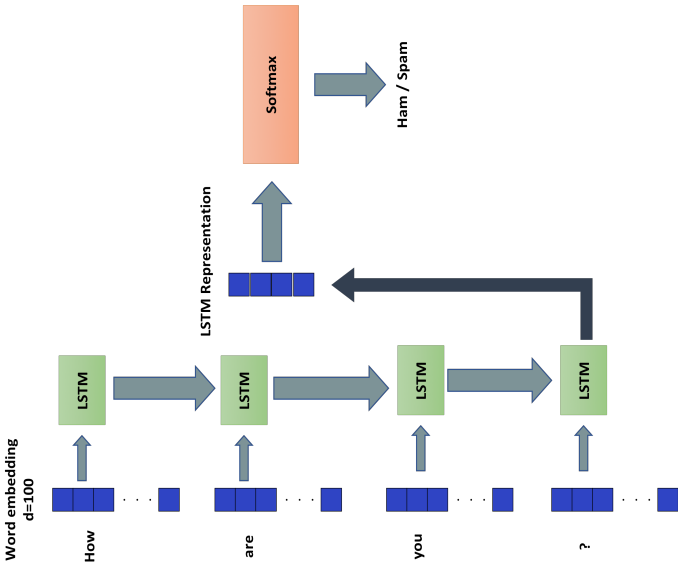


Fig. 2. LSTM Deep Learning Model

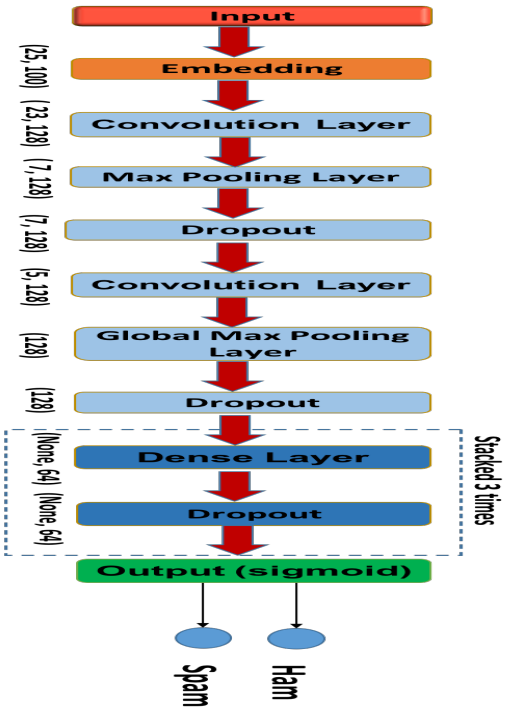


Fig. 3. CNN Deep Learning Model

C. Machine Learning Classifiers

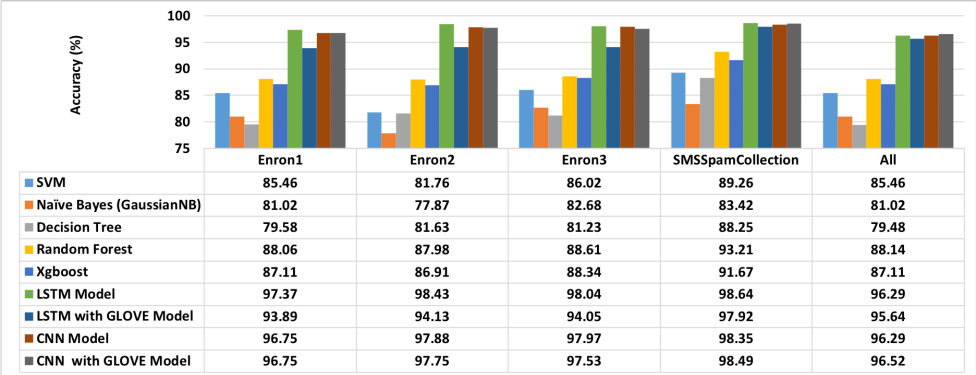
In subsection 3.3.1 we discuss the shallow or traditional machine learning classifiers. In Section 3.3.2. we discuss deep learning classifiers.

1) Traditional Machine Learning: After the pre-processing stage, we extracted occurrence words as features and used TF to select the features. The term frequency (TF) of each word in a document is weight dependent upon the distribution of each word in the files [13]. This represents the importance of each word in the file. These important features are added to a feature matrix which is used for classifying the email into Spam and Not-Spam classes, as shown in Fig. 1 using classifiers such as Random Forest (RF) [14], Support Vector Machine (SVM) [15],Decision Tree [16], Gaussian Naive Bayes (GussianNB) [17] and XGboost (XGB) [18].

2) Deep Learning Classifiers:

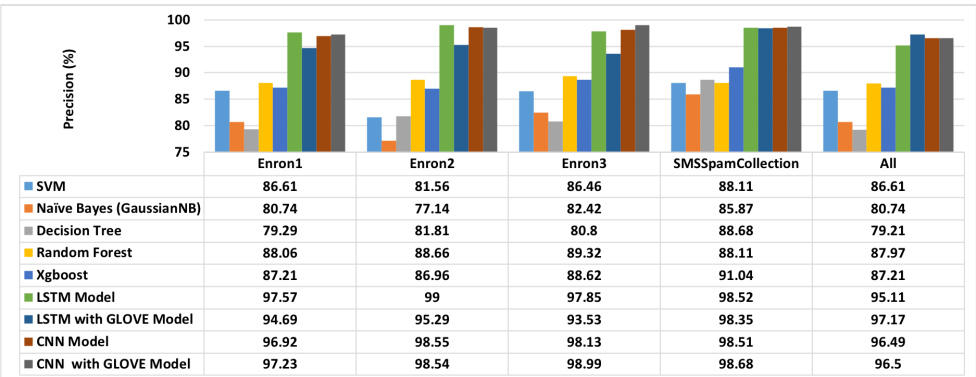
- Long Short-Term Memory (LSTM) Model

The Recurrent Neural Networks (RNN) can learn long-term dependencies. Hochreiter Schmidhuber (1997) introduced these [19]. LSTMs are widely used and operate effectively on large difference problems. All RNNs can sequence repeating



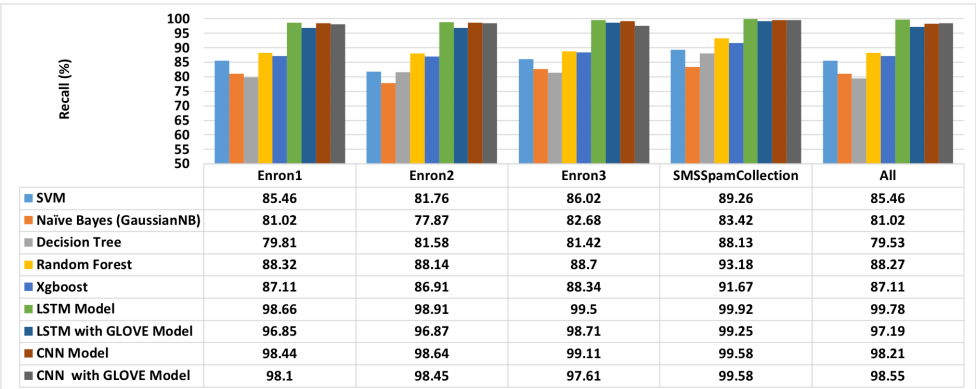
	Enron1	Enron2	Enron3	SMSSpamCollection	All
SVM	85.46	81.76	86.02	89.26	85.46
Naïve Bayes (GaussianNB)	81.02	77.87	82.68	83.42	81.02
Decision Tree	79.58	81.63	81.23	88.25	79.48
Random Forest	88.06	87.98	88.61	93.21	88.14
Xgboost	87.11	86.91	88.34	91.67	87.11
LSTM Model	97.37	98.43	98.04	98.64	96.29
LSTM with GLOVE Model	93.89	94.13	94.05	97.92	95.64
CNN Model	96.75	97.88	97.97	98.35	96.29
CNN with GLOVE Model	96.75	97.75	97.53	98.49	96.52

(a) *Accuracy Assessments*



	Enron1	Enron2	Enron3	SMSSpamCollection	All
SVM	86.61	81.56	86.46	88.11	86.61
Naïve Bayes (GaussianNB)	80.74	77.14	82.42	85.87	80.74
Decision Tree	79.29	81.81	80.8	88.68	79.21
Random Forest	88.06	88.66	89.32	88.11	87.97
Xgboost	87.21	86.96	88.62	91.04	87.21
LSTM Model	97.57	99	97.85	98.52	95.11
LSTM with GLOVE Model	94.69	95.29	93.53	98.35	97.17
CNN Model	96.92	98.55	98.13	98.51	96.49
CNN with GLOVE Model	97.23	98.54	98.99	98.68	96.5

(b) *Precision Assessments*



	Enron1	Enron2	Enron3	SMSSpamCollection	All
SVM	85.46	81.76	86.02	89.26	85.46
Naïve Bayes (GaussianNB)	81.02	77.87	82.68	83.42	81.02
Decision Tree	79.81	81.58	81.42	88.13	79.53
Random Forest	88.32	88.14	88.7	93.18	88.27
Xgboost	87.11	86.91	88.34	91.67	87.11
LSTM Model	98.66	98.91	99.5	99.92	99.78
LSTM with GLOVE Model	96.85	96.87	98.71	99.25	97.19
CNN Model	98.44	98.64	99.11	99.58	98.21
CNN with GLOVE Model	98.1	98.45	97.61	99.58	98.55

(c) *Recall Assessments*

Fig. 4. Performance Comparison of Different Machine and Deep Learning Methods with respect to Different Assessment Metrics on all Enrons and SMS Spam Collection Datasets.

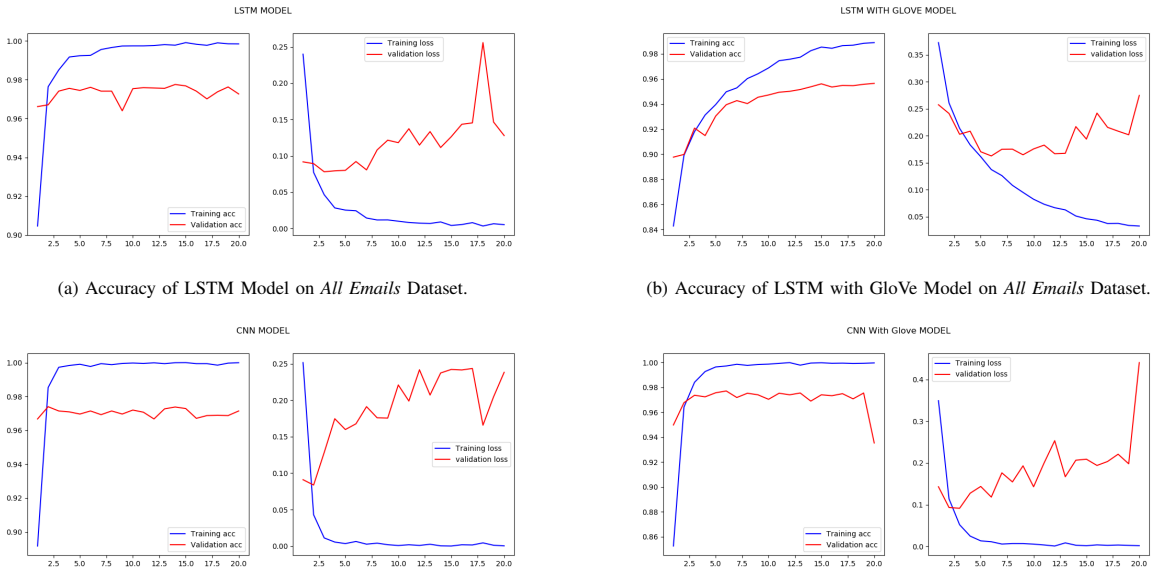
modules of the neural network, for instance, a single tanh layer. The LSTM network can remember the long text sequences [1].

The first layer maps each word to an N-dimensional vector of real numbers and is known as a pre-trained embedding layer. The second layer is an RNN with LSTM units. The final layer is the output layer, with two neurons corresponding to “spam" or “ham" with sigmoid activation functions, as indicated in Fig. 2. We use Global Vector (GloVe) and 100-dimensional vectors. We use *softmaxt* since it provides better results than using *sigmoid* in LSTM with GloVe model. The *softmaxt* function constructs the whole model.

- Convolutional Neural Network (CNN) Model

A Convolutional Neural Network (CNN) has become a popular algorithm in machine learning. A CNN [20], [21] is a neural network where the input is stored in arrays. A CNN has hidden layers, known as convolutional layers, and is used to process 2D arrays of images or audio spectrograms; and for three-dimensional (3D) arrays such as images and videos, pooling layers, and classification layers which is the output

[1]https://easyai.tech/en/ai-definition/lstm/



(a) Accuracy of LSTM Model on *All Emails* Dataset.

(b) Accuracy of LSTM with GloVe Model on *All Emails* Dataset.

(c) Accuracy of CNN Model on *All Emails* Dataset.

(d) Accuracy of CNN with GloVe Model on *All Emails* Dataset.

Fig. 5. Accuracy of All Models on *All Emails* Dataset.

layer and have two neurons each corresponding to “spam" or “ham" with a sigmoid activation function. Because we have applied the CNN model so that it filters spam emails as texts we use an embedding layer with(out) GloVe, as illustrated in Fig. 3, two dimensions convolutional layers *Conv1D*, one max pool *MaxPool1D* layer, four dropout *Dropout* layers, one *GlobalMaxPooling1D* layer, and three *Dense* layers. *Conv1D* layer consists 128 filters, kernel size is 3 and activation method is *relu*. *MaxPool1D* layer size is 2, *Dropout* rate is 0.2, and Dense layer activation method is *relu* and *sigmoid* methods.

D. Data Collection

Every machine learning model requires a training set for training the model. We use open datasets from the community, Enron datasets as shown in Table I. and Enron email as the standard dataset, including spam emails and 10K ham. The Enron company is a bankrupt American firm, once a major player in the energy, commodities, and services industries in the United States [2]. After it had gone bankrupt, the company’s secret emails were distributed across the Internet. In the test phase, we have a 10-folds cross-validation mode. Therefore, we have used the Enron email dataset on both the training and testing phases. We used only email subject and body text. The SMS Spam Collection [3] is a public set of SMS labelled messages collected for research into mobile phone spam. It contains 5,574 ham and spam messages in English as shown in Table I.

IV. Results and Discussion

We use email spam data, including 22,115 messages for each of ham and spam from Enron’s and SMS spam collection datasets. Our baseline uses traditional machine learning algorithms, including five classifiers. We trained and tested each of the classifiers by adding occurrence words as features. Next, we built spam email detection models to compare the impact of shallow machine learning with deep learning models. We performed nine experiments for each of the datasets. Classifier models are produced and tested using cross-validation with 10 folds, using python where to make sure the ratios between spam and not spam classes are identical in each of the folds and the same as in the overall dataset. The results in Fig. 4 show the different machine learning methods of different assessment metrics on all Enron’s and SMS spam collection datasets. Fig. 4 shows the accuracy, recall, and precision of the spam email filtering experiments. The highest accuracy, precision, and recall of the nine classifiers are shown in bold. Deep learning classifiers produce improved results over traditional machine learning classifiers of approximately 10-14%. Random Forest and Xgboost classifiers give better results than the traditional classifiers. The LSTM model is ranked first amongst traditional and deep learning classifiers, however, when all datasets are combined and classifier models built, all deep learning classifiers results are almost identical. The results of all email datasets are provided in Fig. 5a, 5b, 5c and 5d using deep learning algorithms. The figures illustrate the accuracy, loss of training, and validation for all four deep learning classifiers using Embedding word with GloVe or without GloVe. The CNN model is ranked first among deep learning classifiers in regard to accuracy and loss of training and validation curve, such as in Fig. 5c. In Fig. 5a, LSTM mode ranks second in regard to accuracy and loss of training and validation curve.

We use email spam data include 22,115 messages for each of Ham and Spam from Enrons and SMS spam collection datasets. Our baseline uses traditional machine learning algorithms which include five classifiers. We train and test

[2]EnronDatasetofCarnegieMellonUniversity,SchoolofComputerSciencehttps://www.cs.cmu.edu/~enron/

[3]https://github.com/mohitgupta-omg/Kaggle-SMS-Spam-Collection-Dataset-

each classifier by adding occurrence words as features. Then, we build email spam detection models to compare the impact of shallow machine learning with deep learning models. We conduct nine experiments for each dataset. Models of the classifiers are produced and tested using cross-validation with10 folds, using python where we ensure that the ratio between spam and not spam classes is the same in each fold and the same as in the overall dataset. Results for different machine learning methods concerning different assessment metrics on all Enrons and SMS Spam Collection datasets are given in Fig. 4. Fig. 4 shows the accuracy, recall, and precision of email spam filtering experiments. The highest accuracy, precision, and recall of the nine classifiers are shown in bold. Deep learning classifiers produce much better results than traditional machine learning classifiers about 10% to 14%. Random Forest and Xgboost classifiers produce the best results among the traditional classifiers. While the LSTM model ranks first among traditional and deep learning classifiers. But when we combine all datasets and build the models of classifiers, all deep learning classifiers results are almost the same. The results of all email datasets are given in Fig. 5a, 5b, 5c and 5d using deep learning algorithms. These figures show the accuracy and loss of training and validation for all four deep learning classifiers using Embedding word with GloVe or without GloVe. CNN model ranks first among deep learning classifiers based on accuracy and loss of training and validation curve such as Fig. 5c. In Fig. 5a, LSTM mode ranks second based on accuracy and loss of training and validation curve.

V. Conclusion

Email is an inexpensive, effective, and fast way to exchange messages using the internet. Spam email is annoying to end-users, financially damaging, and can be a security risk. The objective of spam email is to collect sensitive personal information about users. The majority of emails in internet traffic contain spam. This work uses deep learning methods, such as CNN and LSTM models with(out) GloVe model, to classify Spam and Not-Spam messages. We have compared our proposed technique to other shallow techniques using machine learning algorithms. The work presented in this paper, based on machine and deep learning algorithms, shows that including more datasets and deep learning models considerably increases the accuracy detection rate, from 85.46% to almost 97.52% after including all datasets (All Emails). Future work can be improved by using a combination of deep learning classifiers based on text and image spams.

References

[1] M. Basavaraju and D. R. Prabhakar, "A novel method of spam mail detection using text based clustering approach," *International Journal of Computer Applications*, vol. 5, no. 4, pp. 15–25, 2010.

[2] G. V. Cormack, *Email spam filtering: A systematic review*. Now Publishers Inc, 2008.

[3] B. Owen and J. Steiner, "Email filtering system and method," Aug. 25 2009, uS Patent 7,580,982.

[4] S. J. Delany, M. Buckley, and D. Greene, "Sms spam filtering: Methods and data," *Expert Sys. with Appl.*, vol. 39, no. 10, pp. 9899–9908, 2012.

[5] E. Blanzieri and A. Bryl, "A survey of learning-based techniques of email spam filtering," *Artificial Intelligence Review*, vol. 29, no. 1, pp. 63–92, 2008.

[6] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Transactions on Neural networks*, vol. 10, no. 5, pp. 1048–1054, 1999.

[7] M. T. Banday and T. R. Jan, "Effectiveness and limitations of statistical spam filters," *arXiv preprint arXiv:0910.2540*, 2009.

[8] A. Radhakrishnan and V. Vaidhehi, "Email classification using machine learning algorithms."

[9] D. Suleiman and G. Al-Naymat, "Sms spam detection using h2o framework," *Procedia computer science*, vol. 113, pp. 154–161, 2017.

[10] V. K. Singh and S. Bhardwaj, "Spam mail detection using classification techniques and global training set," in *Intelligent Computing and Information and Communication*. Springer, 2018, pp. 623–632.

[11] A. D. Kumar, S. KP *et al.*, "Deepimagespam: Deep learning based image spam detection," *arXiv preprint arXiv:1810.03977*, 2018.

[12] G. Jain, M. Sharma, and B. Agarwal, "Spam detection in social media using convolutional and long short term memory neural network," *Annals of Mathematics and Artificial Intelligence*, vol. 85, no. 1, pp. 21–44, 2019.

[13] G. Forman, "Bns feature scaling: an improved representation over tf-idf for svm text classification," in *Proceedings of the 17th ACM conf. on Info. and knowledge management*, 2008, pp. 263–270.

[14] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[15] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[16] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[17] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1995, pp. 338–345.

[18] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *arXiv preprint arXiv:1603.02754*, 2016.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural info. processing sys.*, 1990, pp. 396–404.

[21] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.