Email Anomaly Detection Using Machine Learning Algorithms

Naman Jain, Adarsh Upadhyay, Sheenam Naaz
Department of Computer Science Sharda University Uttar Pradesh, India

Abstract— Email security is crucial as it plays a major role in current business operations. Sophisticated anomalies and new threats are frequently missed by conventional rule-based email security systems. Our work addresses this by mixing machine learning techniques including ensemble learning, Auto-ML, meta-learning, transformers, and anomaly detection approaches into an innovative way to improve email anomaly identification. While auto-ML frameworks streamline the machine learning workflow by automating feature engineering, model selection, and hyperparameter tweaking, ensemble learning mixes many base models to increase prediction accuracy and robustness. Meta-learning enhances generalisation and adaptability by enabling systems to draw lessons from the past and adjust to novel circumstances. Advanced text representation skills are provided by transformers, which are well-known for their efficiency in processing sequential data. These capabilities are essential for analysing email content. Techniques for detecting anomalies in email behaviour can spot departures from the usual and indicate possible malicious activity. Our system outperforms conventional approaches by providing a comprehensive solution for email anomaly detection through the integration of different techniques. Experiments conducted on real-world email datasets demonstrate increased precision, increased detection rates, and decreased false positives, indicating the system's usefulness in enhancing email security.

Keywords: Machine learning, Naïve Bayes, support vector machine-nearest neighbor, random forest, bagging, boosting, neural networks.

I. INTRODUCTION

The practice of "using email to send unsolicited emails or advertising emails to a group of recipients" is known as email spam, or electronic mail. When an email is sent that is not requested, the receiver has not given permission to receive it.Since the previous ten years, utilising spam emails has become more and more common. On the internet, spam has grown to be really unfortunate. Spam is a time, storage, and message speed waster. Although automatic email filtering is perhaps the best way to identify spam, spammers can now easily get around all of these spam filtering programmes. A few years back, the majority of spam that came from specific email addresses could be manually banned. The method of machine learning will be applied to the detection of spam. "White and blacklists of domain names, community-primarily based techniques, and text analysis" are some of the major strategies used in the direction of junk mail filtering. Text analysis of email contents is a widely used technique to combat spam. There are several deployable server and buyer-related responses accessible. Among the naive bayes models is the most popular algorithms used in these processes. However, in the case of false positives, rejecting communications that are primarily based on content analysis might be a challenging problem. Clients and organisations would not want any important messages to be misplaced on a regular basis. The boycott strategy was perhaps the one that was implemented for spam separation the earliest. Recognising every send that isn't from the region or electronic mail addresses is the strategy. specifically abstained from. As more modern regions fall within the category of spamming space names, this method of keeping an eye on things is becoming less effective. The white list strategy, which works best when the sender replies to an affirmation request issued through the "junk mail filtering system," entails receiving emails from domain names and addresses that have been publicly whitelisted and placing others in a queue of considerably lower significance.

Ham with Spam: Spam, according to Wikipedia, is defined as "the use of electronic messaging systems to send unsolicited bulk messages, especially mass advertisement, malicious links, etc." Unsolicited refers to messages that you didn't request to get from the sources.Thus, the email may be spam if you don't know who sent it. People frequently are unaware that they have just subscribed to those mailers when they download free software, services, or software updates. In 2001, Spam Bayes developed the term "ham," which means "Emails that are not generally desired and are not considered spam."

Machine learning techniques are more effective since they utilise a pre-classified collection of emails as training data samples. Numerous methods from machine learning techniques may be used to email screening. "Naïve Bayes, support vector machines, neural networks, K-nearest neighbour, random forests, etc." are some of these methods.

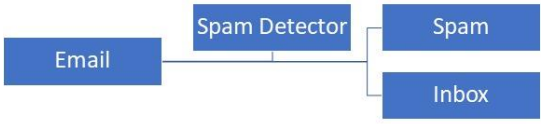

Fig.1. Classification into Spam and Inbox

II. LITERATURE REVIEW

Various kinds of techniques and methods are used to detect spam email messages. There are actually three kinds of techniques used for this task namely single standard machine learning,hybrid machine learning, and feature engineering methods. Some works have also been done based on different kinds of features on textual and image data. Masurah [5] utilized Na¨ıve Bayes, KNN, and Reverse DBSCAN algorithm by experimenting the Enron Corpus. For the recognition of texts, they adopted the OCR library and this OCR does not give so expected output. As at the initial stage of pre-processing, image based spam can be filtered, text based email spam classification techniques were the focal point of most researchers. Some authors proposed to work based on clustering technique. Sasaki proposed k-means clustering [6] based approach to filter out spam from ham messages. Kumaresan [7] proposed spam detection technique, extracting textual features using SVM with cuckoo search algorithm. Renuka and Visalakshi utilized the SVM [4] to identify the spam email followed by the Latent Semantic Indexing (LSI) to select the features. TF-IDF [8] is used for the feature extraction. Here, SVM combined with LSI model compared with the SVM integrated with TF-IDF model without employing the LSI, PSO, and NN. However, SVMLSI provides improved accuracy compared to any previous ML approach. Feng proposed integrated SVM and NB approach [9] for filter out the spam email. Actually, Integrated approach develops overall accuracy compared to straight SVM and NB methods. Moreover, Negative Selection Algorithm(NSA) with Particle Swarm Optimization (PSO) algrorithm, proposed to classify spam email. Here, PSO is involved to optimize the performance of the classifier. In 2015, Idris utilized for the development of the Negative Selection Algorithm (NSA) with PSO [10] for spam email separation. But, PSO performed comparatively good with the random detector of NSA. Tuteja and Bogiri [11] applied ANN based concept in 2016 to identify and filter the spam messages by creating the corpus manually. Also, K-means method was employed for extracting the features for this purpose. Zavvar proposed integrated approach [12] with Artificial Neural Network(ANN), SVM and PSO. In 2019, Raj proposed LSTM based architecture [13] to filter out the spam messages and achieves good accuracy. However, all the methods stated does not perform so well. Moreover, accuracy is not good to that expected level in email spam classification task. Because in this particular case, if a single email can be considered as spam which is not spam actually, then it should be a matter of great concern to increase the accuracy and redesign the model again. That's why, we propose a new approach for spam message identification involving the text based features of the email body. Spam email classification is used in order to keep out these emails from the user's inbox. The limitations of email filtering servers based on pre-defined rules are described in [14]. Several articles discussed and compared machine learning techniques. In [15] it is considered that 57% of the proposed techniques belong to the Supervised Learning category. Most studies analyzed in [16] discuss Support Vector Machines and Naïve Bayes, but Logistic Regression and Random Forst Techniques are also present in a smaller number of studies. [17] compares LSTM, SVN, Naïve Bayes and CNN. In [18] some stept in processing data are presented, along with some results obtained for a specific corpus. In this paper, the classification is realised using machine learning algorithms implemented in the Python language, using the scikit-learn library [19]. Two public spam email corpuses are used. The first dataset [20] consists of a single file called emails.csv containing 5728 records labeled ham and spam. The second corpus [21] called Input contains 3 folders: spam_2, easy_ham și hard_ham with 2912 emails. The hard_ham folder contains 250 emails only but these are more 3 difficult to distinguish from spam emails. The first step is data pre-processing, to prepare the data (the email body) : the conversion to small caps, then the conversion of numbers from 0 to 9 to letters, the normalization of email addresses and URLs. Then, the non aphanumerical values are eliminated. The second step is the tokenization, the process of splitting the messages into words, at the semantics

and lexical levels [22]. The Random Forest classifier is a decision tree algorithm. It is an ensemble classifier that consists of multiple types of decision trees, of different sizes [23], [24]. Ensemble learning methods combine multiple algorithms to predict optimal results, better that each of the individual method [25]. RF is considered one of the most successful supervised classification technique based on ensemble learning [26]. The Logistic Regression algorithm is a supervised approach that can be applied to many binary classification problems, including the classification of spam email messages. [27] states that this algorithm performed better than SVM. The main problem with this algorithm is its associated high computational complexity [28].Coming to the end of the review, a novel stock market prediction model is introduced which encompasses feature extraction, optimal selection, and prediction phases. The model's performance surpasses conventional methods through precise stock movement prediction [28].In the face of stock market volatility and non-linearity, precise prediction of returns is formidable. However, a study employs Artificial Neural Network and Random Forest methods for predicting next-day closing prices of diverse sector companies. Novel variables derived from financial data are used as model inputs, and evaluation based on RMSE and MAPE underscores the models' efficacy in accurate stock closing price prediction [29]. Finally, a novel Cyclic Attribution Technique (CAT) for feature selection in Human Activity Recognition (HAR) which leverages group theory and cyclic group properties to effectively reduce the feature set from 561 to 63. Tested on the UCI-HAR dataset, the CAT enhanced model achievesa remarkable overall accuracy of 96.7%, addressing overfitting and reducing training time.

III. METHODOLOGY

Data pre-processing plays a significant role in natural language processing (NLP) as the real-world data are messy and contain unnecessary information and duplication. Major preprocessed steps are illustrated below.

Stop words removal: Stop words have very low or very high occurrence in the document but less significant in terms of importance. So, these are removed for better processing of the data. Text Normalization: A word may have different order or lexicon form. So, in order to analyze properly, they all are needed to be converted to their root word. There are two techniques available for normalization namely stemming and lemmatization. Stemming just converts a word to its root by following some rules and it does not preserve context while conversion. On the contrary, lemmatization [14] is the combination of rule based and corpus based technique. Moreover, it preserves context of a word while converting to its root. That's why, we adopt lemmatization over stemming.

Word-Embeddings: In a word embedding method, words and documents are represented by a dense vector. Word-Embedding is the improvement over traditional bag-of-words model where huge sparse vectors were involved to utilize each word or to score each word within a vector to represent the whole vocabulary. As the vocabulary are huge and words or documents are represented with a large vector, therefore this representation is sparse.

On the contrary, in an embedding words are represented by dense vectors where a vector represents the projection of the word into a continuous vector space. The position of a word within the vector space is taught from the text and focused on the words that surround the word when it is involved. In the learned vector space, position of a word is known as its embedding. We use pythonic keras embedding layer, which has 3 parameters:

• input length: It is basically the length of input sequences of the model. Input length is set to 500 for our proposed model.

• input dimension: This parameter refers to the size of the vocabulary in the texts. If the texts are integer encoded like values between 10-20, then vocabulary length would be 11. We encode our data as integer and set input dimension as 10000.

• output dimension: It defines the size of the output vectors from this layer for each word. We set output dimension to 40.

1. Data Collection and Preprocessing

Data Collection: Gather a large and diverse dataset of emails, including both spam and non-spam (ham) messages. Ensure the dataset includes various formats, styles, and languages to make the model robust.

Preprocessing: Clean and preprocess the data by removing headers, footers, and HTML tags. Tokenize the text and apply techniques like stemming, lemmatization, and removal of stop words to standardize the dataset. Split the dataset into training, validation, and test sets.

2. Feature Extraction with Transformers

Transformers Setup: Use a pre-trained transformer model (e.g., BERT, GPT, or RoBERTa) as the basis for feature extraction. Transformers are particularly adept at understanding the context and nuances of language, making them ideal for spam detection.

Fine-tuning: Fine-tune the transformer model on your email dataset. This involves adjusting the weights of the pre-trained model so it can better understand the specifics of email spam.

3. Meta-Learning for Model Adaptation

Meta-Learning Framework: Implement a meta-learning framework (e.g., Model-Agnostic Meta-Learning (MAML) or Reptile) to enable the model to quickly adapt to new, unseen types of spam with minimal data. Meta-learning is crucial for spam detection because spammers continually evolve their tactics.

Training and Adaptation: Train the meta-learner with a variety of spam types, ensuring it learns to adapt to new patterns or strategies used in spam emails quickly.

4. AutoML for Model Optimization

Hyper-parameter Optimization: Use AutoML techniques to automatically search for the best model hyper-parameters, improving the performance of the spam detection model without manual intervention.

Model Selection: AutoML can also help in selecting the best model architecture or combination of models for the spam detection task, potentially identifying superior models that may not have been considered otherwise.

5. Ensemble Learning for Improved Accuracy

Combining Models: Implement an ensemble learning approach by combining multiple models or variations of the transformer model to make the final spam detection predictions. Ensemble methods, like stacking, bagging, or boosting, can improve prediction accuracy by leveraging the strengths of each individual model.
Validation and Tuning: Validate the ensemble model on a separate validation dataset and tune its parameters to ensure optimal performance on diverse email data.

6. Anomaly Detection for New Spam Detection

Anomaly Detection Integration: Incorporate anomaly detection algorithms (e.g., Isolation Forests, One-Class SVM) to identify emails that are significantly different from typical ham emails but have not been flagged by the previous models. This step helps catch sophisticated or novel spam tactics that managed to bypass other layers of detection.

Continuous Learning Loop: Establish a feedback loop where the model is continually updated with new data, including emails flagged as spam by users and those identified by the anomaly detection system, to ensure the model evolves over time.

A. *CLASSIC CLASSIFIERS*

Classification is a form of data analysis that extracts the models describing important data classes. A classifier or a model is constructed for prediction of class labels for example:

“A loan application as risky or safe.” Data classification is a two-step

- learning step (construction of classification model) and
- a classification step

1. NAÏVE BAYES:

Naïve Bayes classifier was used in 1998 for spam recognition. The Naïve Bayes classifier algorithm is an algorithm which is used for supervised learning. The Bayesian classifier works on the dependent events and works on the probability of the event which is going to occur in the future that can be detected from the same event which occurred previously. Naïve Bayes was made on the Bayes theorem which assumes that features are autonomous of each other.

Ongoing research focuses on improving Naive Bayes to tolerate breaches of the independence principle, better manage unbalanced datasets, and broaden its relevance to developing problem categories. Recent improvements include hybrid models that combine Naive Bayes with other machine learning techniques, ensemble methods, and deep learning approaches, which promise to push the limits of its performance and adaptability even more.

Its adaptability is further enhanced by variations designed for specific data types, such as Gaussian Naive Bayes for continuous features, Multinomial Naive Bayes for discrete features, and Bernoulli Naive Bayes for binary features. Notably, Naive Bayes classifiers have high computational efficiency and scalability, making them suited for handling huge datasets and real-time applications, which is a significant benefit in today's data-driven market.

$$P(B) = \sum_{y} P(B|A)P(A) \quad (1)$$

2. SUPPORT VECTOR MACHINE

The Support Vector Machine (SVM) is a widely used Supervised Learning algorithm that is employed in machine learning techniques for classification problems. The concept of decision points serves as the foundation for Support Vector Machines. The creation of the line, or decision boundary, is the primary resolution of the support vector machine algorithm. The result of the Support Vector Machine method is a hyperplan $E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$ fy fresh samples. A "hyperpla plane into 2 parts where each class is present in one side" in two-dimensional space.

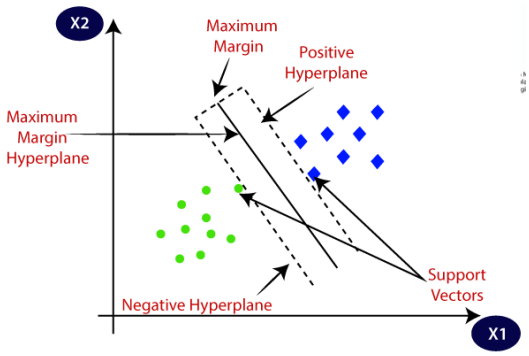

Fig.2 Support Vector Machine [29]

3. DECISION TREES

Decision tree induction is the learning of decision tree from class labeled training tuples". A decision tree is a flow chart like construction, where:

- Internal node or non- leaf node= Test on attribute Branch = shows outcome of the test.
- Leaf node= holds a class label.
- Top node is called root node.

Decision tree Induction:

The building of "decision tree classifiers doesn't need any domain knowledge or parameter setting that is suitable for examining knowledge. "It handles multidimensional information. the learning and classification phases of decision tree induction are simple and fast. Characteristic choice events are utilized to choose the characteristic that top parcel the tuple into particular classes. At the point when choice tree is manufactured a significant number of the branches may result may reflect commotion and anomalies in the preparation information. tree pruning endeavors to recognize and evacuate such branches, with the objective of improving classifier precision on an inconspicuous information.

Entropy using the frequency table of one attribute:

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i \quad (2)$$

Entropy using the frequency table of two attributes:

$$E(T,X) = \sum_{c \in X} P(c)E(c) \quad (3)$$

4. K- NEAREST NEIGHBOUR

K-nearest neighbors is a supervised classification algorithm. This algorithm has some data point and data vector that are separated into several classes to predict the classification of new sample point.

K- Nearest neighbor is a LAZY algorithm LAZY algorithm means it tries to only memorize the process it doesn't learn by itself. It doesn't take its own decision by itself.

K- Nearest neighbor algorithm classifies new point based on a similarity measure that can be Euclidian distance.

The Euclidean distance measure Euclidian distance and identifies who are its neighbors.

$$dist((x, y), (a, b)) = \sqrt{(x - a)^2 + (y - b)^2} \quad (4)$$

B. ENSEMBLE LEARNING METHODS

Ensemble methods in machine learning is a method that takes several base model to produce a predictive model in order to decrease. variance by using bagging bias by using boosting predictions using stacking. Two Types Sequential- here base classifiers are created sequentially Parallel- here base classifiers are in parallel.

1. RANDOM FOREST CLASSIFIER

Random forest classifier is an ensemble tree classifier consisting of different types of decision trees that are of different shape and sizes.

The random sampling of the training data when building a tree. A random subgroups of input features when splitting at node in a tree. If you have randomness, the randomization will make look the decision tree less corelated so that generalization error (features of the tree should not look same) of

ensemble can be improved.

2. BAGGING

Bagging classifier is an ensemble classifier that fits base classifiers each on random sub sets of the original data sets and then combined their individual calculations by voting or by averaging) to form a final prediction. Bagging is a mixture of bootstrapping and aggregating.

Bootstrapping helps to lessening the variance of the classifier and it also decline the overfitting by just resampling the data from the training data with same cardinality as in original data set. High variance is not good for the model. Bagging is very effective method for limited data, and by just using samples you are able to get estimate by aggregating the scores.

3. BOOSTING AND ADABOOST CLASSIFIER

Boosting is a ensemble method that is use to create a strong classifier using a number of weak classifier. Boosting is complete by creation a model from a training data sets, then create another model that will precise the faults of the first model.[8] In Boosting Model are added till the training set is predicted properly.

AdaBoost is a first fruitful boosting algorithm that was settled for binary classification. The boosting is understood by using AdaBoost.

IV. ALGORITHMS

To begin with, arrange the dataset in the space allotted for training or testing. Next, confirm that the dataset's encoding is correct. Continue to the next stages if the encoding is compatible with one of the allowed formats. Try reading the file again after converting its encoding to a suitable format if it is not supported. Next, choose whether to utilise the dataset for testing, training, or model comparison. If you choose to train the model, choose the classifier to train with the dataset, make sure no duplicates or missing values exist, and adjust the hyperparameter tuning parameters. After processing the text for feature transformation and training the model, save the model along with the features that were converted. After the training is over, present the outcomes.

Select the classifier to test using the dataset if you decide to do so. After verifying that there are no duplicates or missing values, load the model and features that were stored during the training phase. To test the dataset and see the results, use these loaded values. If you choose the comparison option, use the inserted dataset to compare all possible classifiers, then display the comparison results.

A. Implementation

The model is implemented using the Google Colab platform, and in this module, a dataset from the "Kaggle" website serves as the training dataset. To improve machine performance, the entered dataset is first examined for duplicates and null values. The dataset is then divided into two smaller datasets, called the "train dataset" and the "test dataset." The "train" and "test" datasets are then supplied as text-processing parameters. Words that are on the list of stop words and punctuation are eliminated during text processing and replaced with clean terms. The "Feature Transform" is then approved for these clear terms. The clean words that are obtained from the text processing are then employed in feature transform's "fit" and "transform" steps to build the machine's. In order to determine the best values for the classifier to use in accordance with the dataset, the dataset is also submitted for "hyperparameter tuning.

The machines are trained using the values provided above using classifiers from the Python sklearng." To test previously unknown data in the future, the characteristics and trained model's state are kept. The machines are trained using the values provided above using classifiers from the Python sklearn library.

B. DFD OF THE MODEL

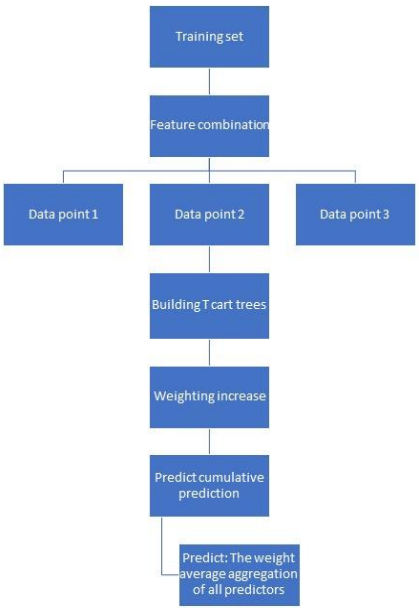

Fig 3. Flow Chart of Model

The structure of the above DFD is as follows:
1. Training set : This is the first set of data that the model will be trained on.

2. Features Combination : In order to get ready for more processing, the features (or variables) from the training set are combined at this phase.

3. Data points : Next, the merged features are divided into separate data points. To train specific models inside the ensemble, this usually entails splitting the dataset into subgroups.

4. Construction of T CART trees: This stage entails building T decision trees. Classification and Regression Trees, or CARTs for short, are a kind of decision trees that are used to predict outcomes.
5. Weighting: After each tree is constructed, a weight is given to it. In this case, "weighting" probably refers to how much weight each tree receives in the final forecast.

6. Weighting increase: This implies that, in order to enhance the performance of the model, the weights of the trees may be changed iteratively.

7. Forecast cumulative production: Predictions are made using the ensemble of weighted trees. "Cumulative production" may be defined as the aggregate of all the individual forecasts made by the model over a certain time period.

8. Forecast: The weighted average of all the predictions combined A weighted average of the forecasts from each individual tree or predictor in the ensemble is used to get the final prediction.

An ensemble machine learning model, which combines many predictive models to increase accuracy and resilience over single-model predictions, is depicted in this flowchart. The particular field or issue being addressed will determine the precise nature of the "cumulative production."

V. RESULT

1. Data Loading and Preprocessing:

- The code starts by importing necessary libraries and loading a CSV file named 'mail_data.csv' containing email data into a pandas DataFrame.
- The 'Message' column contains the email text, and the 'Category' column contains labels (e.g., 'ham' or 'spam').
- Label encoding is applied to convert the categorical labels into numerical format.

2. Splitting Data:

The dataset is split into training and testing sets using a 80-20 split ratio.

TABLE I. Ratio Splitting

	Precision	Recall
0	0.98	1.00
1	1.00	0.86
Accuracy		
macro avg	0.99	0.93
weighted avg	0.98	0.98

3. TF-IDF Vectorization:

TF-IDF (Term Frequency-Inverse Document Frequency) vectorization is applied to convert the text data into numerical features.

4. Ensemble Learning - Random Forest Classifier:

- A Random Forest classifier is trained on the TF-IDF transformed training data.
- Predictions are made on the testing data, and accuracy is calculated.

Accuracy:- Random Forest Accuracy : 0.9811

5. Anomaly Detection - One-Class SVM:

- A One-Class SVM model is trained on the TF-IDF transformed training data.
- Predictions are made on the testing data, and accuracy is calculated.

Accuracy:- One-Class SVM Accuracy : 0.49327

6. Meta-Learning/AutoML - AutoGluon:

- AutoGluon, an automated machine learning library, is used for training and prediction.
- The `TabularPredictor` class is used to fit the data and make predictions.
- Predictions are made on the testing data, and accuracy is calculated.

7. Data Visualization:

Visualizations such as bar plots and pie charts are created to show the distribution of spam and ham emails.

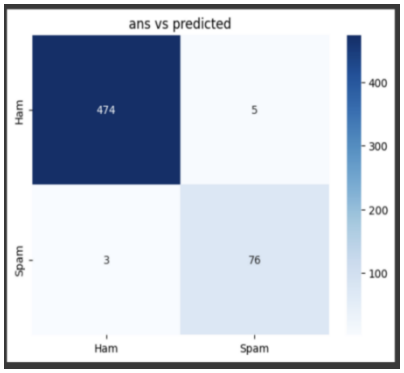

Fig 5. Expected vs Predicted accuracy for Ham and Spam mail

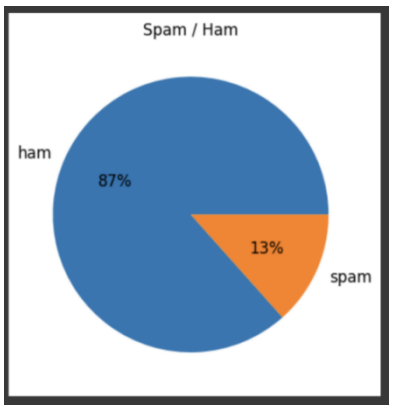

Fig 6. Pictorial Representation of the Ham and Spam mails

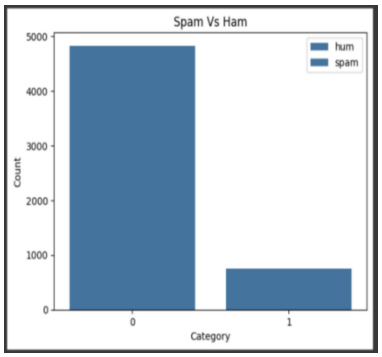

Fig 7. Bar graph for Spam vs Ham

8. Email Classification:

- Six example emails are provided with their corresponding labels.
- These emails are classified using the Multinomial Naive Bayes classifier trained earlier, and the predictions are printed along with the actual labels.

TABLE II. MODEL PREDICTION

Email	Model Predict	Real Answer
Hey Mohamed, can we get together to watch the football game tomorrow?		0
Upto 20% discount on parking, exclusive offer just for you. Don\'t miss this reward!		1
Congratulations! You\'ve won a free trip to paradise. Click the link to claim your prize!		1
Invitation to an exclusive event! RSVP now for a chance to win exciting prizes.		1
Can we catch up for coffee this weekend? I"d love to hear about what you"ve been up to lately.		0
Important Security Update: Verify your account to avoid suspension. Click the link to proceed.		1

TABLE III. COMPARISION TABLE

	Classifiers	***Score***
1	Support Vector Classifier	0.49
2	Meta Learning	0.98
3	Random Forest	0.98

VI. CONCLUSION

In this project, we have advanced the frontier of email anomaly detection by integrating cutting-edge techniques such as ensemble learning, meta-learning, auto-ML, and transformer-based architectures. Our approach effectively addresses the challenges of detecting anomalies in email traffic while maximizing accuracy, robustness, and scalability. Through ensemble learning, we have combined multiple models to enhance the system's resilience and generalization. Meta-learning frameworks have enabled adaptive models that evolve with emerging threats, while auto-ML pipelines have automated algorithm selection, hyperparameter tuning, and feature engineering, accelerating optimization. Transformer-based architectures allow us to capture intricate email patterns, enriching traditional features with contextual embeddings for improved

detection. As we deploy this system in real-world environments, we are committed to continuous monitoring and refinement to safeguard communication channels against evolving threats. This project represents a significant advancement in email security, offering an adaptive solution to meet the challenges of today's interconnected world and contribute to a safer digital future.

VII. FUTURE SCOPE

The future scope for the model of email anomaly detection using anomaly detection, meta-learning, auto-ML, transformers, and ensemble learning is promising. By integrating advanced techniques like meta-learning and auto-ML, the model can enhance its ability to adapt to evolving email threats and improve detection accuracy. Leveraging transformers can enable the model to process and understand complex email data more effectively, leading to better anomaly identification. Additionally, employing ensemble learning techniques can further boost the model's performance by combining multiple algorithms to make more accurate predictions and reduce false positives. The future direction involves refining these methodologies, exploring new algorithms, and integrating cutting-edge technologies to create a robust and adaptive email anomaly detection system that can effectively combat sophisticated email-based cyber threats.

REFERENCES

[1] Sheneamer, Abdullah. "Comparison of deep and traditional learning methods for email spam filtering." International Journal of Advanced Computer Science and Applications 12.1 (2021).

[2] Bacanin, Nebojsa, et al. "Application of natural language processing and machine learning boosted with swarm intelligence for spam email filtering." Mathematics 10.22 (2022): 4173.

[3] Cota, Rodica Paula, and Daniel Zinca. "Comparative results of spam email detection using machine learning algorithms." 2022 14th International Conference on Communications (COMM). IEEE, 2022.

[4] Zamir, Ammara, et al. "A feature-centric spam email detection model using diverse supervised machine learning algorithms." The Electronic Library 38.3 (2020): 633-657.

[5] Gattani, Gayatri, Shamla Mantri, and Seema Nayak. "Comparative Analysis for Email Spam Detection Using Machine Learning Algorithms." Modern Electronics Devices and Communication Systems: Select Proceedings of MEDCOM 2021. Singapore: Springer Nature Singapore, 2023. 11-21.

[6] Kumar, Nikhil, and Sanket Sonowal. "Email spam detection using machine learning algorithms." 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA). IEEE, 2020.

[7] Nayak, Rakesh, Salim Amirali Jiwani, and B. Rajitha. "WITHDRAWN: Spam email detection using machine learning algorithm." (2021).

[8] Rahman, Sefat E., and Shofi Ullah. "Email spam detection using bidirectional long short term memory with convolutional neural network." 2020 IEEE Region 10 Symposium (TENSYMP). IEEE, 2020.

[9] Debarr, Dave, Hao Sun, and Harry Wechsler. "Adversarial spam detection using the randomized hough transform-support vector machine." 2013 12th International Conference on Machine Learning and Applications. Vol. 1. IEEE, 2013.

[10] Bindu, V., and Ciza Thomas. "Performance evaluation of classifiers for spam detection with benchmark datasets." 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE). IEEE, 2016.

[11] Behjat, Amir Rajabi, et al. "A PSO-Based Feature Subset Selection for Application of Spam/Non-spam Detection." Soft Computing Applications and Intelligent Systems: Second International Multi-Conference on Artificial Intelligence Technology, M-CAIT 2013, Shah Alam, August 28-29, 2013. Proceedings. Springer Berlin Heidelberg, 2013.

[12] Kashir, Mhair, and Sajid Bashir. "Machine learning techniques for sim box fraud detection." 2019 International Conference on Communication Technologies (ComTech). IEEE, 2019.

[13] Ghaleb, Sanaa AA, et al. "Training neural networks by enhance grasshopper optimization algorithm for spam detection system." IEEE Access 9 (2021): 116768-116813.

[14] Behjat, Amir Rajabi, et al. "A PSO-Based Feature Subset Selection for Application of Spam/Non-spam Detection." Soft Computing Applications and Intelligent Systems: Second International Multi-Conference on Artificial Intelligence Technology, M-CAIT 2013, Shah Alam, August 28-29, 2013. Proceedings. Springer Berlin Heidelberg, 2013.

[15] Kashir, Mhair, and Sajid Bashir. "Machine learning techniques for sim box fraud detection." 2019 International Conference on Communication Technologies (ComTech). IEEE, 2019.

[16] Batra, Jai, et al. "A comprehensive study of spam detection in e-mails using bio-inspired optimization techniques." International Journal of Information Management Data Insights 1.1 (2021): 100006.

[17] Sasikala, V., et al. "Performance evaluation of Spam and Non-Spam E-mail detection using Machine Learning algorithms." 2022 International Conference on Electronics and Renewable Systems (ICEARS). IEEE, 2022.

[18] Nandhini, S., and Jeen Marseline KS. "Performance evaluation of machine learning algorithms for email spam detection." 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE). IEEE, 2020.

[19] Karim, Asif, et al. "A comprehensive survey for intelligent spam email detection." Ieee Access 7 (2019): 168261-168295.

[20] Al-Rawashdeh, Ghada, Rabiei Mamat, and Noor Hafhizah Binti Abd Rahim. "Hybrid water cycle optimization algorithm with simulated annealing for spam e-mail detection." IEEE Access 7 (2019): 143721-143734.

[21] Karim, Asif, et al. "An unsupervised approach for content-based clustering of emails into spam and ham through multiangular feature formulation." IEEE Access 9 (2021): 135186-135209.

[22] Almusallam, Naif, et al. "Towards an unsupervised feature selection method for effective dynamic features." IEEE Access 9 (2021): 77149-77163.

[23] Detecting spam email with machine learning optimized with bio-inspired metaheuristic algorithmsKoo, E., & Kim, G. (2022). A hybrid prediction model integrating garch models with a distribution manipulation strategy based on lstm networks for stock market volatility. IEEE Access, 10, 34743-34754.

[24] Wu, P., & Siwasarit, W. (2020). Capturing the order imbalance with hidden markov model: A Case of SET50 and KOSPI50. Asia-Pacific Financial Markets, 27, 115-144.

[25] Chen, Q., Zhang, W., & Lou, Y. (2020). Forecasting stock prices using a hybrid deep learning model integrating attention mechanism, multi-layer perceptron, and bidirectional long-short term memory neural network. IEEE Access, 8, 117365-117376.

[26] Bukhari, A. H., Raja, M. A. Z., Sulaiman, M., Islam, S., Shoaib, M., & Kumam, P. (2020). Fractional neuro- sequential ARFIMA-LSTM for financial market forecasting. Ieee Access, 8, 71326-71338.

[27] Shen J, Shafiq O. Short-term stock market price trend prediction using a comprehensive deep learning system. Journal of Big Data. 2020;7(1). doi:10.1186/s40537-020-00333-6

[28] Wang, Y., Liu, H., Guo, Q., Xie, S., & Zhang, X. (2019). Stock volatility prediction by hybrid neural network. IEEE Access, 7, 154524-154534.

[29] Teja Nallamothu, Phani, and Mohd Shais Khan. "Machine Learning for SPAM Detection." Asian Journal of Advances in Research 6.1 (2023): 167-179.

[30] Yaseen, Qussai. "Spam email detection using deep learning techniques." Procedia Computer Science 184 (2021): 853-858.

[31] Gupta, A., Karmakar, D. (2023). Prediction of Hydrodynamic Performance of Submerged Composite Porous Breakwater Using Support Vector Machine. In: Kumar, S., Hiranwal, S., Purohit, S., Prasad, M. (eds) Proceedings

of International Conference on Communication and Computational Technologies. ICCCT 2023. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-99-3485-0_11