OPTIMIZING HYPERPARAMETERS FOR ENHANCED EMAIL CLASSIFICATION AND FORENSIC ANALYSIS WITH STACKED AUTOENCODERS

Merly Thomas[1] and B. B. Meshram[2]

[1]Department of Computer Engineering, Fr. Conceicao Rodrigues College of Engineering, Bandra, Mumbai
[2]Veermata Jijabai Technological Institute Mumbai, India

ABSTRACT

Electronic mail, commonly known as email, is a crucial technology that enables streamlined operations and communications in corporate environments. Empowering swift and dependable transactions, email is a driving force behind heightened productivity and organizational effectiveness. However, its versatility also renders it susceptible to misuse by cybercriminals engaging in activities such as hacking, spoofing, phishing, email bombing, whaling, and spamming. As a result, effective and efficient data analysis is important in avoiding and detecting cyber-attacks and crime on times. To overcome the above challenges, a novel approach named Aquila Optimization (AO) is used in this paper to find the best set of hyperparameters of the Stacked Auto Encoder (SAE) classifier. The purpose of increasing the hyperparameters of the SAE using the AO is to obtain a higher text classification accuracy. Then the optimized SAE classifies the selected features into different classes. The experimental results showed that the proposed AO-SAE model outperforms the existing models such as Logistic Regression (LR) and Long Short-Term Model based Gated Current Unit (LSTM based GRU) in terms of Accuracy.

KEYWORDS

Aquila Optimization, Cybercrimes, Email forensic dataset, ReliefF algorithm, Stacked Auto Encoder

1. INTRODUCTION

Email communication has grown considerably in the latest eras because of its low cost, ease, and rapidity [1], [2], [3], [4]. It is mainly employed in business, educational, technical discussions, and file transactions. It enables non-intrusive communication with individuals around the world. E-mail is a popular means of communication, but it is also utilized by hackers to perform crimes. E-mails are used to commit cybercrimes such as hacking, spoofing, phishing [5], [6], E-mail bombing, whaling, as well as spamming [7]. Spam or bulk e-mail has now emerged as a major problem on the Internet which is a significant and widespread attack that involves sending unwanted messages, malware, as well as phishing via email to affected computers [8], [9]. A new email spam analysis discovered that about 14.5 billion emails are created in a day, worldwide. About 2.5% of these emails are labeled malicious emails [10]. Fake links are inserted in the content of emails, causing consumers to be sent to false Sites. The false URLs in this operation replicate well-known Web sites, making them stranger [11], [12]. Moreover, sending and receiving a significant amount of spam emails generate congestion in the network and delays. Technically, blocking spam communications would keep the network from collapsing. Identifying and confirming actual emails would enhance email security and assist in the protection of user resources [13]. Whereas human spam identification is possible but filtering out a significant quantity of spam emails may be time-consuming & costly [14]. In machine learning

or deep learning, the body of the email is used to determine whether an email is spam or not. To overcome the limitations, several studies are conducted using ML and DL approaches which are used to detect and classify spam emails with different processes. However, different types of issues such as misclassification, low accuracy, and high classification error occur during the implementation phase. In this paper, a novel Aquila Optimization approach is used to find the best hyperparameters set of Stacked Auto Encoder to improve the text classification accuracy which is detailed in the following sections. The main contribution that is included in this research is given as follows:

- In this research, the email forensic dataset is employed which is a publicly available open-source dataset that comprises various kinds of emails.
- In the pre-processing stage, the input data is pre-processed using various kinds of pre-processing techniques such as Tokenization, Stemming and Lemmatization for reducing multiple forms of the word to one form.
- Next, the feature extraction and feature selection processes are performed using their respective techniques. Following that, for initializing the Stacked Auto encoder the hyper-parameter optimization is conducted using the proposed AO by selecting the best set of hyperparameters.
- Finally, using the Stacked Auto-encoder the classification process is performed and the results are the e-mail classes as normal, harassment, fraudulent, and suspicious.

This research paper is organized as follows: The related works on spam email classification are presented in Section 2. A detailed explanation of the proposed methodology is given in Section 3. Section 4 presents the outcomes of the proposed method whereas the conclusion is presented in Section 5.

2. LITERATURE SURVEY

Maryam Hina et al. [15] suggested a multi-label email classification system to manage emails. The process begins with mail information obtained from the Enron email database, which has four groups. The dataset was unbalanced, but we manually balanced it to guarantee that the model training made fair selections. The dataset is divided into four categories: fraudulent, harassing, typical, and unusual. The initial dataset has three classifications; we included another to group all emails as one class. The optimal parameters are found utilizing a grid-search method and 10-fold cross-validation across the characteristics listed in the parameter estimation table. Nevertheless, it is a time-consuming & exhausting method that required a big amount of email content for effective analysis.

Maryam Hina et al. [16] suggested SeFACED, a unique effective method for multiclass email classification that employs a Gated Recurrent Unit (GRU) relying on Long Short-Term Memory (LSTM). SeFACED concentrates on modifying LSTM-based GRU parameters to get the greatest performance as well as evaluation by contrasting it to classical machine learning, deep learning models, as well as cutting-edge research in the field. The highest E-mail size has been more than 1000 words, requiring the use of several sequence modules; popular sequence learning methods include the LSTM & GRU. As a result, the LSTM + GRU has better accuracy for the test data. Although sampling approaches can overcome the issue of data imbalance, they influence the model's efficiency.

B. Aruna Kumara [17] suggested an improved data pre-processing technique for multi-category email categorization. REVA University's Internal Quality Assurance Cell (IQAC) validated the research dataset. The term "sustainability" refers to the process of creating a sustainable lifestyle.

The datasets were divided as samples for training and testing in an 8:2 ratio. A detailed accuracy investigation revealed that the suggested method enhances the accuracy of every ML classifier. Whenever compared to big datasets, several classifiers demonstrated improved accuracy. Whereas if email content sign choices include graphics, the suggested model doesn't eliminate them because the study is primarily focused on text categorization systems.

Khalid Iqbal et al. [18] suggested an innovative ML technique for spam email detection. A spambase UCI dataset of around 5000 cases was used to reduce the possibilities of overfitting. When implementing the ML model, methods for feature selection were used to pre-process the information to enhance the accuracy model. The 10-fold approach was used to evaluate the model. The accuracy was improved by adopting Point-Biserial feature selection, which allows everyone to extract the important characteristics for spam email categorization. To achieve optimal results, the ANN is used in the UCI spambase email dataset, however, the feature selection approach is not employed in the suggested model to choose the optimal features from the data so it may have an impact on classification accuracy.

Akhilesh Kumar Shrivas et al. [19] proposed a robust text classifier for categorizing scam email text using a feature selection method. The study involved gathering six different types of Enron datasets, combining them into seven final Enron datasets. The researchers employed the WEKA data mining tool to analyze these datasets after pre-processing, which included removing unnecessary terms. The SymmetricalUncert FST merged the Enron datasets before classification and analysis using an RF technique, demonstrating superior accuracy with smaller feature subsets. However, it's worth noting that the suggested FST removed characteristics with values much less than a threshold, including the elimination of short meaningful sentences.

3. PROPOSED METHOD

In this section, the entire process included in this research is briefly explained and the flowchart of the proposed method is depicted in Figure 1.

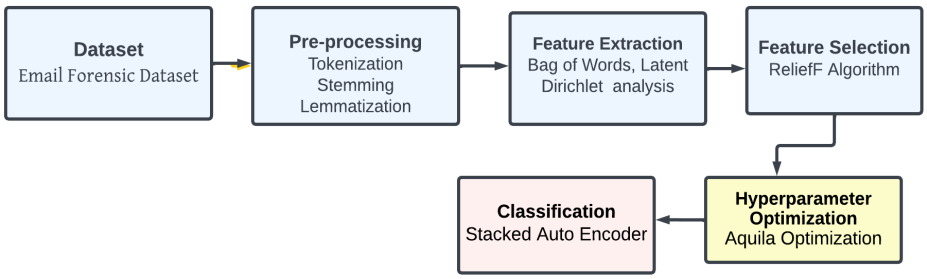

Figure 1: Flow chart of the proposed methodology

3.1. Dataset

In this research, the four kinds of email forensic datasets known as Enron, Phished e-mails corpora, Hate Speech, and Offensive datasets are used. Where the Enron dataset, a large-scale email collection from a real organization that contains many normal emails, and fraudulent emails are provided by the phished emails corpora. Similarly, the Hate Speech and Offensive dataset comprises harassment messages, threat messages, terrorism messages, etc. The collected dataset is given as input to the following procedures.

3.2. Pre-processing

The input data is sourced from a publicly available email forensic dataset. Text pre-processing is a crucial phase in Text Classification (TC) and text mining, involving techniques like Tokenization, Stemming, and Lemmatization, frequently explored in both ML & DL.

- **Tokenization**: Tokenization divides emails into tokens, which can be phrases, word groups, links, or sections, separated by intervals like white spaces, special characters, semi-colons, commas, etc.
- **Stemming**: The technique of decreasing words to their base, word stem, is called stemming. Words like seeing, sees, or seen, for example, are simplified to see. A few of the terms may not be viable in the language sometimes.
- **Lemmatization:** Lemmatization appropriately reduces conjugated words, ensuring the underlying word aligns with the language, relying on lexicon and word form. The goal is to eliminate inflectional ends and restore words to their base or dictionary form. Pre-processed data is then inputted for the feature extraction process to extract features from the processed data.

3.3. Feature Extraction

The process of feature extraction is performed using the pre-processed data from the prior stage. Here, for extracting the features from the pre-processed data various feature extraction techniques such as Bag of Words (BoW), Latent Dirichlet Analysis (LDA), and Term Frequency-Inverse Document Frequency (TF-IDF) are used. The aforementioned techniques are briefly explained in the following;

- **Bag of Words (BoW):** It is a common as well as simple approach for extracting features from textual information. It is a method of extracting textual information for use in modeling, which includes ML techniques. A BoW is a textual representation that depicts the repetition of words inside a source. It entails two steps. (1) A known-word lexicon; (2) an indicator of the prevalence of known-words.
- **Latent Dirichlet Analysis (LDA):** It is a more advanced version of the hierarchical Bayesian model. The main idea is to consider a document as a set of words, every document as a mixture of many concepts, and every topic as a set of various words. The LDA has the benefits of supervised learning, variable extension, significantly enhanced computation speed, & proven efficacy.
- **Term Frequency-Inverse Document Frequency (TF-IDF):** It represents one of the most frequently employed weighing measures for determining the link between words as well as documents. It has been used to extract word features for text classification and other NLP applications. Words with greater TF-IDF weights are considered more relevant and are retained, whereas words with lesser weights are considered less relevant and are removed. Lastly, the extracted features are employed in the feature selection procedure to choose the best features.

3.4. Feature Selection

Selecting the optimal features using a best approach for performing smooth classification process is known as feature selection. In this research, best features are selected from the extracted features using a ReliefF algorithm. The ReliefF technique is employed for working with multi-class challenges. Every time, the ReliefF method selects a random sample $R \in \mathbb{R}^d$ from the training dataset D. Choosing R's k-nearest neighbours H_j, (j = 1,2,..., k) from samples within the

similar class as R, and determining R's k-nearest neighbours $M_j(C)$, (j = 1,2,..., k) of R derived from samples of a distinct class than R, where Euclidean distance is employed to discover the KNN. The above procedure is performed m times. The weight of every feature is then modified using Equation (1), with the difference computed by Equation (2). As a result, feature selection is carried out based on the weight of every feature as well as the set threshold.

$$w(f_i) = w(f_i) - \sum_{j=1}^{k} \frac{diff(f_i, R, H_j)}{mk} + \sum_{C \neq class(R)} \frac{p(C)}{1 - p(classclass(R))} \times \sum_{j=1}^{k} \frac{diff(f_i, R, M_j(C))}{mk}, (i = 0, 1, \ldots, d), \quad (1)$$

$$diff(A, R_1, R_2) = \begin{cases} \dfrac{|R_1[A] - R_2[A]|}{\max A - \min A}, & if\ A\ is\ continuous \\ 0, & if\ A\ is\ discrete\ and\ R_1[A] = R_2[A], \\ 1, & if\ A\ is\ discrete\ and\ R_1[A] \neq R_2[A], \end{cases} \quad (2)$$

Where, $diff\ (A, R_1, R_2)$ denotes the variance in samples R_1 & R_2 on feature A, $R_1[A]$ & $R_2[A]$ signify the values of samples $R_1$and R_2 on feature A, while max A and min A express the highest and lowest values of every sample on feature A [20].

3.5. Proposed Hyperparameter Optimization

The major goal of hyperparameter optimization is to improve text classification performance by increasing the hyperparameters of the Stacked Auto Encoder classifier. Optimizing hyperparameters is an important aspect of regulating the learning behavior of the developed models. If the hyperparameters are not properly tuned, the developed model parameters yield unsatisfactory results since they do not minimize the loss function. So, a hyperparameter optimization is used for obtaining the best classification outcomes. In this work, for hyperparameter optimization, the Aquila Optimizer (AO) is utilized. AO is a revolutionary meta-heuristic optimization technique influenced by Aquila's natural behaviour while prey capture. AO was created to optimize real-world parameters as well as functionalities Where the hunting approaches for slow-moving prey represent the method's local exploitation ability. The AO algorithm has a high global exploration capability, a high search efficiency, and a quick convergence time which are used to optimize the hyperparameter of the SAE classifiers. The following parameters & their limits are presented in this research Dropout [0.1-0.4], Learning Rate [0.003-0.1], L2Regularization [0.003-0.1], and Max-Epoch [5,10,15,20]. The AO method begins with the initial solutions, which are produced at random, then repeatedly tries to increase the text classification model's accuracy till stopping conditions are reached. The fitness function consists of Stacked Auto Encoder networks that execute the evaluation & deliver the accuracy of text categorization. Moreover, in this paper, accuracy is used as a fitness function to achieve the best values of Hyperparameters.

The below steps reveal the search processes of the AO approach:

1. **Initialization process:** every optimization procedure begins with random values of possible solutions (X_{ij}) in the range within the upper (up_j) & lower (lo_j) bounds, which is described in the following equation (3):

$$X_{ij} = rand \times (up_j - lp_j) + lp_j) \quad (3)$$

Where,
$rand$ - random integer,
ith - amount of populations,
jth - issue dimension size.

2. Initially, the Aquila investigates any prey in the search space. This exploration procedure is carried out at a high level, which is known as enlarged exploration in the search space. As the Aquila detects prey, it lowers with a vertical stoop to acquire it. Statistically, this behavior is expressed in the following equation (4):

$$X_1(t+1) = X_b(t) \times \left(1 - \left(\frac{t}{T}\right)\right) + X_M(t) - X_b(t) \times rand), \quad (4)$$

Where, $X_1(t+1)$ - Aquila's positioning in the subsequent iteration of t,

$X_b(t)$ - optimum solution acquired up to the tth iteration.
$1 - \left(\frac{t}{T}\right)$ - utilized to regulate the search space exploration procedure,
$X_M(t)$ - mean value of the solutions from the prior iteration, as computed by Equation (5).

$$X_M(t) = \frac{1}{N}\sum_{I=1}^{n} X_i(t), \quad (5)$$

Where, N is the size of the population.

3. The second method, which Aquila prefers, is searching at contour f lying with a brief glide attack. As a result, the Aquila is near to the chased prey, resulting in confined exploration of the search field. This procedure is represented mathematically in equation (6):

$$X_2(t+1) = X_b(t) \times Levy(D) + X_R(t) + (y - x) \times rand, \quad (6)$$

Where, $X_R(t)$- random solution obtained at the ith iteration,

$Levy(D)$ - dimension space (D) levy flight distribution function, which is computed using the equation (7):

$$Levy(D) = s \times \frac{u \times \omega}{|v|^{1/\beta}} \quad (7)$$

Where, s - constant value equivalent to 0.01,
u & v - random numbers among 0 and 1.
ω - dynamic adaptive coefficients computed as described in the following equation (8):

$$\omega = \left(\frac{T(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{T\left(\frac{1+\beta}{2} \times 2^{\left(\frac{\beta-1}{2}\right)}\right)}\right) \quad (8)$$

Where, β- constant equivalent to one 1.5. The exploration procedure has a spiral form that is managed by x & y values, which is calculated using the following equations (9), (10), (11), (12), and (13):

$$y = r \times \cos(\theta) \quad (9)$$
$$x = r \times \sin(\theta) \quad (10)$$
$$r = r_1 + U \times D_1 \quad (11)$$
$$\theta = -W \times D_1 + \theta_1 \quad (12)$$
$$\theta_1 = 3 \times \frac{\pi}{2} \quad (13)$$

Where,

r_1 - value from 1 to 20,
U – value equivalent to 0.00565,
D_1 - integer number ranging from 1 to the search space size,
W – value equivalent to 0.005.

4. The third procedure is increased exploitation (X_3), which is relevant to every flooring prey with a slower escape reaction. Aquila is traveling at a low altitude, and after it has chosen its target, it captures it gradually with a slow descending approach. This phase is statistically represented by the equation (14):

$$X_3(t+1) = \left(X_b(t)\right) \times \alpha - ramd + (rand \times (up - lp) + lp) \times \delta \quad (14)$$

Where, α & δ - the exploitation modification variables, which have values ranging from 0.1 to 0.9 and are set at 0.1 based on experiments for many benchmarks.

5. The fourth procedure (X_4) happens like whenever the Aquila travels on the ground & collects prey. It is considered a narrower exploitation phase when dealing with huge prey. This stage is statistically represented in Equation (15):

$$X_4(t+1) = QF \times X_b(t) - (G_1 \times X(t) \times rand) - G_2 \times Levy(D) + rand \times G_1 \quad (15)$$

Where, QF - quality function utilized to balance the search techniques, and Equation (16) is used to compute it.

G_1- multiple AO movements created by Equation (17).
G_2 - AO's flight slope, which decreases from 2 to 0 as the prey measure the flow rate from first to final location and is represented by Equation (18).

$$QF(t) = t^{\frac{2 \times rand() - 1}{(1-T)^2}} \quad (16)$$
$$G_1 = 2 \times rand() - 1 \quad (17)$$
$$G_2 = 2 \times \left(1 - \frac{t}{T}\right) \quad (18)$$

The specific pseudocode is shown in Algorithm 1.

Algorithm 1 Aquila Optimizer

1. Input = Hyper parameter ranges with feature data
2. Initialization phase
3. Initialize the solutions X and algorithm parameters.
4. while (The end condition is not met) do
5. Calculate the fitness values.
6. Determine the best solution.
7. for (i=1, to N) do
8. Update the $X_M(t), x, y, G_1, G_2, Lecy\ (D)$, etc.
9. if $t \leq \left(\frac{2}{3}\right) * T$ then
10. if $rand \leq 0.5$ then
11. Update the current solution using Equation (4).
12. else
13. Update the current solution using Equation (6).
14. end if
15. else
16. if $rand \leq 0.5$ then
17. Update the current solution using Equation (14).
18. else
19. Update the current solution using Equation (15).
20. end if
21. end if
22. end for
23. end while
24. return The best solution (X_b).
25. Output = Best hyper parameter set

3.6. Classification

In this research, text classification is carried out using a Stacked Auto-Encoder (SAE) once the best feature vectors have been chosen. SAE is a feed-forward NN with one or more hidden layers its primary goal is to recreate the input data unsupervised. It is made up of an encoder, which converts the input data into low-dimensional forms, and a decoder, which recreates the actual data from the encoder output. With an autoencoder, the amount of output nodes equals the amount of input nodes. The possibility of missing values while text categorization is small with a Stacked auto-encoder.

4. EXPERIMENTAL RESULTS

In this research, the analysis and classification of emails are performed using the email forensic dataset to reduce the malicious or fraudulent attacks produced by hackers. The Aquila Optimization algorithm is proposed in this research to increase the hyperparameters of the classifier named SAE. For the precise classification, the deep learning-based SAE classifier is employed. The performance of the feature selection algorithm (ReliefF algorithm), optimization algorithm (Aquila Optimization), as well as classifier (SAE), is evaluated using the common performance measures such as Accuracy, Sensitivity, Specificity, F1-score as well as Matthew's correlation coefficient (MCC). The obtained results are compared with various feature selection algorithms, classifiers, and optimization algorithms. The mathematical equation for the performance measures is given in Table 1.

Table 1: Mathematical equation of respective performance measures

Performance Measures	Equations
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Sensitivity	$\frac{TP}{TP+FN}$
Specificity	$\frac{TN}{TN+FP}$
F1-score	$\frac{2*Precision*Recall}{Precision+Recall}$
MCC	$\frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

4.1. Performance Analysis

Here, Table 2 represents the performance analysis of the employed ReliefF feature selection algorithm with existing feature selection algorithms such as Infinite Feature Selection (IFS), Infinite Latent Feature Selection (ILFS). Table 2 results that the ReliefF algorithm achieves a higher accuracy of 98%, sensitivity of 97.74%, specificity of 97.49%, f1-score of 97.11% and MCC of 97.36%. Whereas the IFS, ILFS, and Relief achieve accuracy of 91.22%, 94.63%, 95.85% respectively.

Table 2: Performance evaluation of ReliefF algorithm with existing algorithms

Algorithms	Accuracy (%)	Sensitivity (%)	Specificity (%)	F1-Score (%)	MCC (%)
IFS	91.22	90	90.36	87.63	89.29
ILFS	94.63	93.73	93	93.90	93
Relief	95.85	94	94.5	94	93.45
ReliefF	**98**	**97.74**	**97.49**	**97.11**	**97.36**

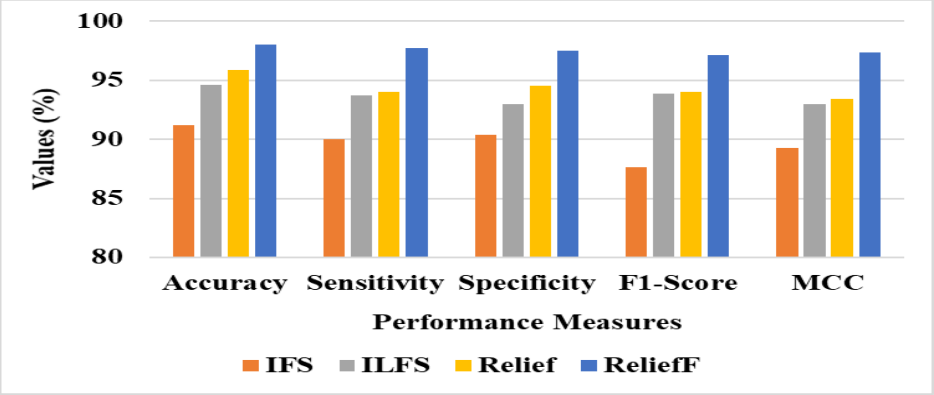

Figure 2: Graphical depiction of ReliefF algorithm vs Existing algorithms

The performance outcomes are graphically depicted in Figure 2. It demonstrates that the employed ReliefF algorithm outperforms the existing feature selection approaches such as IFS, ILFS, Relief in terms of Accuracy, Sensitivity, Specificity, F1-score as well as MCC. Whereas the performance evaluation of AO algorithm with existing algorithms is given in Table 3.

Table 3: Performance evaluation of Aquila Optimization algorithm with existing algorithms

Algorithms	Accuracy %)	Sensitivity (%)	Specificity (%)	F1-Score (%)	MCC (%)
PSO	94	91.93	93	92.83	93.34
GWO	95.12	93	94	94.74	94.70
ABC	96	94.3	94.23	95	95.65
Mayfly	96.63	95	95.36	96.33	96
AO	**98**	**97.74**	**97.49**	**97.11**	**97.36**

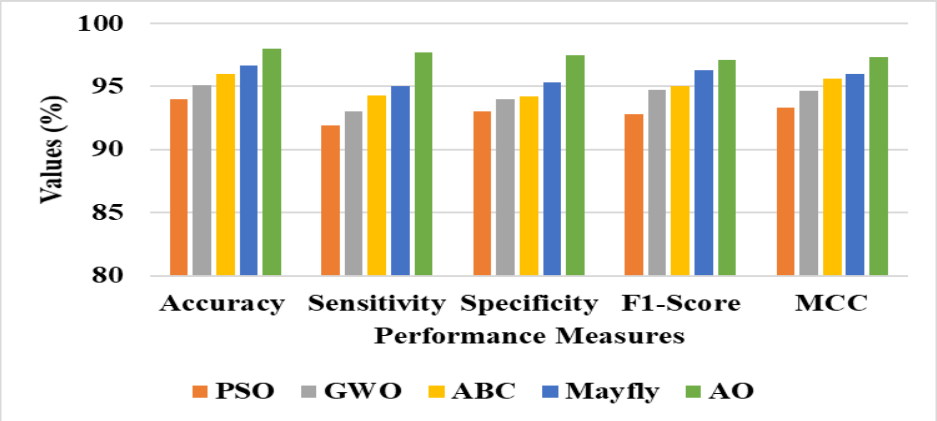

Figure 3: Graphical Representation of Aquila Optimizer vs existing optimization algorithm

Table 3 results that the AO algorithm achieves a higher accuracy of 98%, sensitivity of 97.74%, specificity of 97.49%, f1-score of 97.11% and MCC of 97.36%. Whereas the PSO, GWO, ABC, Mayfly achieves accuracy of 94%, 95.12%, 96% and 96.63% respectively. The graphical representation of AO with the existing optimization algorithm is depicted in Figure 3. It demonstrates that the employed AO algorithm outperforms the existing optimization algorithm such as Particle Swarm Optimization (PSO), Grey Wolf Optimization (GWO), Artificial Bee Colony algorithm (ABC), Mayfly algorithm in terms of Accuracy, Sensitivity, Specificity, F1-score as well as MCC. Whereas the performance evaluation of SAE classifier with existing classifiers such as Generative Adversarial Networks (GAN), Sparse autoencoder, Recurrent Neural Networks (RNN), and Convolutional Neural Network (CNN) is given in Table 4.

Table 4: Performance evaluation of SAE classifier with existing classifiers

Classifiers	Accuracy (%)	Sensitivity (%)	Specificity (%)	F1-Score (%)	MCC (%)
GAN	96	95.83	95.40	95	94
Sparse AE	96.17	95.65	95.53	95.30	94.89
RNN	96.85	96	95.14	96	95.32
CNN	97.39	96.52	96.75	96	96.12
SAE	**98**	**97.74**	**97.49**	**97.11**	**97.36**

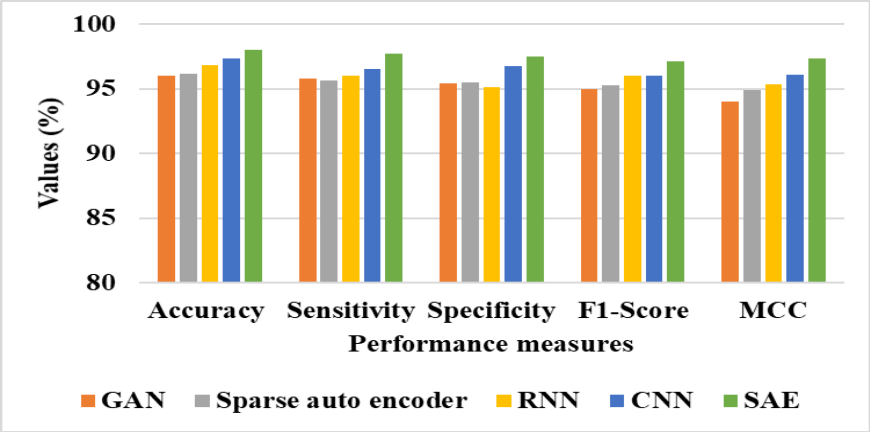

Figure 4: Graphical depiction of SAE classifier vs Existing Classifiers

Table 4 shows that the SAE classifier achieves a higher accuracy of 98%, sensitivity of 97.74, specificity of 97.49, f1-score of 97.11 and MCC of 97.36. Whereas the GAN, Sparse auto encoder, RNN, CNN achieves accuracy of 96%, 96.17, 96.85%, 97.39% respectively. The graphical depiction of SAE classifier with the existing classifier is depicted in Figure 4. It demonstrates that the employed AO algorithm outperforms the existing optimization algorithm such as GAN, Sparse autoencoder, RNN, CNN, Mayfly in terms of Accuracy, Sensitivity, Specificity, F1-score as well as MCC.

4.2. Comparative Analysis

This section provides a comparative analysis of the proposed Aquila Optimization based Stacked Auto Encoder (AO-SAE) model with existing models, such as LR [15] and LSTM based GRU [16], which are used to evaluate the performance of the AO-SAE. Then the proposed AO-SAE is compared and analysed with the existing models in terms of classification accuracy. In Table 5 the AO-SAE is compared to the LR [15] and LSTM based GRU [16].

Table 5: Comparison evaluation of the proposed model with existing models

Models	Classification Accuracy (%)
LR [15]	91.91
LSTM based GRU [16]	95
AO-SAE	**98**

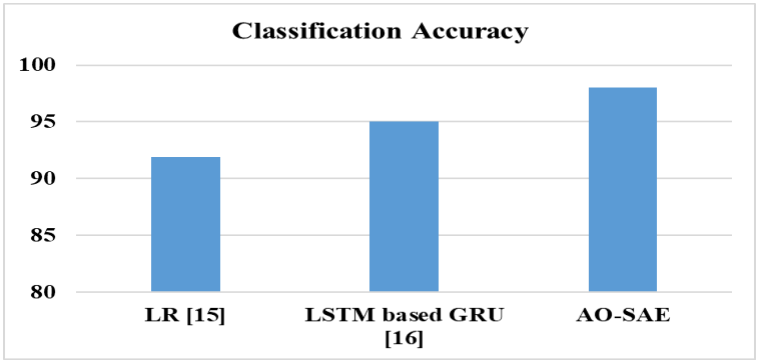

Figure 5: Graphical depiction of the proposed AO-SAE model with existing models

From Table 5, the proposed AO-SAE model achieves a greater accuracy of 98%, and the compared LR [15] and LSTM based GRU [16] models achieve 91.91% and 95% respectively. The comparison between the proposed AO-SAE with existing models is graphically depicted in Figure 5. As a result, the proposed AO-SAE achieves greater classification accuracy and it clearly states that the proposed AO-SAE model outperforms the existing LR [15] and LSTM based GRU [16] models.

5. CONCLUSION

In this paper, to identify the best hyperparameters set of Stacked Autoencoder (SAE), a novel Aquila Optimization (AO) is proposed for higher text classification accuracy. The data is collected from the Email forensic dataset which is a publicly available dataset used in the entire process. Next using pre-processing techniques such as Tokenization, Stemming and Lemmatization the input data is smoothened. Then the pre-processed data is transferred to perform the process the feature extraction where the features the extracted using the BoW as well as Latent Dirichlet Analysis (LDA) techniques. Later, using the ReliefF feature selection algorithm, the respective process is conducted where the optimal features are selected to perform precise classification. Finally, the popular SAE classifier is employed to classify the selected optimal features. To evaluate the performance of the proposed AO approach common performance measures such as Accuracy, Sensitivity, Specificity, MCC as well as F1-Score are used. The experimental results show that the proposed AO-SAE model obtains a greater accuracy of 98%, which outperforms the other two compared existing approaches such as LR model and LSTM based GRU model.

REFERENCES

[1] Hosseinalipour, Ali, and Reza Ghanbarzadeh. "A novel approach for spam detection using horse herd optimization algorithm." Neural Computing and Applications 34, no. 15 (2022): 13091-13105.

[2] Jánez-Martino, Francisco, Eduardo Fidalgo, Santiago González-Martínez, and Javier Velasco-Mata. "Classification of spam emails through hierarchical clustering and supervised learning." arXiv preprint arXiv:2005.08773 (2020).

[3] Mrisho, Zubeda K., Jema David Ndibwile, and Anael Elkana Sam. "Low Time Complexity Model for Email Spam Detection using Logistic Regression." International Journal of Advanced Computer Science and Applications 12, no. 12 (2021).

[4] Prosun, Priyo Ranjan Kundu, Kazi Saeed Alam, and Shovan Bhowmik. "Improved Spam Email Filtering Architecture Using Several Feature Extraction Techniques." In Proceedings of the International Conference on Big Data, IoT, and Machine Learning: BIM 2021, pp. 665-675. Springer Singapore, 2022.

[5] Kumar, Abhishek, Jyotir Moy Chatterjee, and Vicente García Díaz. "A novel hybrid approach of SVM combined with NLP and probabilistic neural network for email phishing." International Journal of Electrical and Computer Engineering 10, no. 1 (2020): 486.

[6] Rathee, Dhruv, and Suman Mann. "Detection of E-mail phishing attacks–using machine learning and deep learning." International Journal of Computer Applications 183, no. 1 (2022)

[7] Sumathi, P., R. Elavarasan, M. Sadhakrishnan, and K. Harishanand. "E-mail Sink AI: Deep Learning model for multiclass E-mail Classification for Forensic Analysis using Design Thinking." Journal of Positive School Psychology (2022): 5425-5432.

[8] Reddy, A., Maheswari, M. U., Viswanathan, A., & Vikram, G. (2022). Using Support Vector Machine For Classification And Feature Extraction Of Spam In Email. International Journal of Innovation in Engineering, 2(2), 26–32. https://doi.org/10.52547/ijie.2.2.26

[9] Mustafa, Mustafa, Imam Riadi, and Rusydi Umar. "Header investigation for spam email forensics using framework of national institute of standards and technology." ILKOM Jurnal Ilmiah 13, no. 2 (2021): 163-167.

[10] Muralidharan, Trivikram, and Nir Nissim. "Improving malicious email detection through novel designated deep-learning architectures utilizing entire email." Neural Networks 157 (2023): 257-279.

[11] Magdy, Safaa, Yasmine Abouelseoud, and Mervat Mikhail. "Efficient spam and phishing emails filtering based on deep learning." Computer Networks 206 (2022): 108826.
[12] Samarthrao, Kadam Vikas, and Vandana M. Rohokale. "Enhancement of email spam detection using improved deep learning algorithms for cyber security." Journal of Computer Security 30, no. 2 (2022): 231-264.
[13] Hadi, Suha Mohammed, Ali Hakem Alsaeedi, Dhiah Al-Shammary, Zaid Abdi Alkareem Alyasseri, Mazin Abed Mohammed, Karrar Hameed Abdulkareem, Riyadh Rahef Nuiaa, and Mustafa Musa Jaber. "Trigonometric words ranking model for spam message classification." IET Networks (2022).
[14] Mageshkumar, N., A. Vijayaraj, N. Arunpriya, and A. Sangeetha. "Efficient spam filtering through intelligent text modification detection using machine learning." Materials Today: Proceedings 64 (2022): 848-858.
[15] Hina, Maryam, Mohsan Ali, Abdul Rehman Javed, Gautam Srivastava, Thippa Reddy Gadekallu, and Zunera Jalil. "Email classification and forensics analysis using machine learning." In 2021 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI), pp. 630-635. IEEE, 2021.
[16] Hina, Maryam, Mohsin Ali, Abdul Rehman Javed, Fahad Ghabban, Liaqat Ali Khan, and Zunera Jalil. "Sefaced: Semantic-based forensic analysis and classification of e-mail data using deep learning." IEEE Access 9 (2021): 98398-98411.
[17] Kumara, B. Aruna, Mallikarjun M. Kodabagi, Tanupriya Choudhury, and Jung-Sup Um. "Improved email classification through enhanced data preprocessing approach." Spatial Information Research 29 (2021): 247-255.
[18] Iqbal, Khalid, and Muhammad Shehrayar Khan. "Email classification analysis using machine learning techniques." Applied Computing and Informatics ahead-of-print (2022).
[19] Shrivas, Akhilesh Kumar, Amit Kumar Dewangan, and Samrendra Mohan Ghosh. "Robust Text Classifier for Classification of Spam E-Mail Documents with Feature Selection Technique." Ingénierie des Systèmes d'Information 26, no. 5 (2021).
[20] Zhang, Baoshuang, Yanying Li, and Zheng Chai. "A novel random multi-subspace based ReliefF for feature selection." Knowledge-Based Systems 252 (2022): 109400.Lee, S.hyun. & Kim Mi Na, (2008) “This is my paper”, ABC *Transactions on ECE*, Vol. 10, No. 5, pp120-122.