**International Journal of Research Publication and Reviews**

Journal homepage: www.ijrpr.com ISSN 2582-7421

# A New Deep Learning Network Detecting Hidden Messages in Digital Image Steganalysis

***A. M Yola[1], Usman Ali A[2], A. Y. Gital[2], K. M. Musa[3], Hajara Musa[3]***

[1]ICT Directorate, Federal University of Kashere Gombe, Nigeria

[2,] Department of Computer Sciences, Federal collage of Education Technical Gombe, Nigeria

[2, 3] Department of Mathematical Sciences, Abubakar Tafawa Balewa University Bauchi, Nigeria

[3]Department of Computer Sciences, Gombe State University, Gombe, Nigeria

**ABSTRACT**

Steganalysis is detecting and analysing hidden information in digital materials, often hidden using steganography. Steganalysis has recently received a great deal of attention from both the Information Security Unit and the media, especially the Information and Communication tool. In this paper, the study systematically digital image steganalysis using profound learning and tries to analyze the current stage of cutting-edge image steganalysis structures on deep learning. Existing activation functions like ELU, ReLU, and Leaky ReLU have limitations due to computational cost and dead neurons. To cover this limitation, A new hybrid activation function is proposed, using a convolutional neural network (CNN) plus a Hybrid activation function to simplify feature extraction and classification procedures. The proposed algorithm outperforms the existing activation functions with81% accuracy.

Lastly, procedural tasks of the approaches and numerous encouraging tips on profound learning image steganalysis suggest the need to demonstrate the way such encounters could be transformed into productive impending research opportunities.

***KEYWORDS**: Steganalysis, Digital Image Steganalysis, Deep Learning, Convolutional Neural Network, Activation Function*

## 1.0 INTRODUCTION

Steganalysis is the process of detecting hidden messages within digital media, such as images, sounds, videos, and plain-text documents(Wang et al., 2020). The use of steganography involves concealing a message within a "Transporter/Cover media" to establish secret messages. Steganalysis is the speciality of identifying the presence of concealed or hidden messages, and it involves isolating stego objects and non-stego objects without knowledge of steganography-based techniques or algorithms(You et al., 2021). The steganalysis approach is classified based on the steganographic algorithm, including specific or target, blind or universal. The tool used for hiding messages in different types of records is the Steghide command-line tool, which deploys robust encryption algorithms to ensure the security of the hidden messages(Ranjan & Forensics, 2016). Another tool, OpenStack, is a free and open-source steganography tool that can work with JPEG and PNG image formats. It offers various encryption and compression algorithms to hide messages(Ravi & Alazab, 2021). Additionally, there are steganography algorithms like COVER, TEST, JUNIWARD, and UERD(Yan et al., 2021). These algorithms were developed for digital image steganalysis, which detects hidden information in digital images(Qian et al., 2017).JMiPOD (JPEG Minimum Embedding) is a steganography algorithm that embeds secret data within the compressed data of JPEG images.Universal embedding with an ensemble of reversible data-hiding techniques is a steganography algorithm that embeds secret data within the pixel values of digital images(B. Chen et al., 2020). It uses a combination of highly reversible data-hiding techniques to embed the secret message, making it more robust against detection by steganalysis algorithms. For safe message hiding, it offers encryption and compression options. For the safety of hiding messages, 256-bit Advanced Encryption Standard (AES) encryption is used(Atta & Ghanbari, 2021). Statistical techniques, such as spatial and transform domains, are used in steganalysis, but they lack the capability of providing information on embedding algorithms(Chutani & Goyal, 2019). Machine learning and deep learning techniques, such as CNN, RNN, DNN and others, have been introduced and applied to overcome this challenge inside the image steganalysis framework. Controlled learning involves classification(Aggarwal, 2018) while unsupervised learning addresses regression(H. Li et al., 2016). Both feature extraction and classification play essential roles in these processes.Since we are dealing with complexity, the optimization process and Data complexity are proposed to look into in other to determine the hyperparameter and dimensionality of data plus the number of class levels. Consequently, deep learning enhances image steganalysis, leading to improved results.

In this research, we focus on using CNN as our model process plus hybridized activation functions to show improvement against the existing works. The proposed approach simplifies feature extraction and classification procedures while potentially improving their efficacy by uniting them into a cohesive end-to-end framework. This led to the advancement of steganography with the hope of improving its functionalities and performance.

## 2.0 LITERATURE REVIEW

### *2.1 The Concept of Steganalysis and Steganography*

As mentioned, earlier, steganalysis involves detecting the existence of a concealed message in an image. Image steganalysis is the process of locating data hidden in cover images through steganography. Image steganalysis deals with a cover or stego image(M. Chen et al., 2017). The concept of digital image steganalysis techniques according to deep learning does not separate the image texture intricacy, to combine images for training purposes(Fridrich & Kodovsky, 2012)(Eid et al., 2022). In general, steganalysis is classed into visible steganalysis, structural steganalysis, statistical steganalysis and gaining knowledge of steganalysis. Visual and structural steganalysis is ideal for guide inspection, and visible steganalysis relies upon a particular construal of visible data. Statistical and gaining knowledge of steganalysis is ideal for automatic calculation(Hussain et al., 2020). Consequently, the distinction between image content with dissimilar texture intricacies is higher than the variances produced by steganographic indicators, which is uncomplimentary for extracting effective steganalysis features(Tan & Li, 2014). The concept of steganalysis is only complete when steganography is included. In addition, feature-based steganalysis approach ideas are conceived for selecting and absorbing appropriate information from cover images(Signoroni et al., 2019).

### *2.1.2 Spatial Domain Approach*

For about a decade, an Ensemble Classifier trained with rich format structures has been the leading method in the identification of furtive messages concealed in an image(Yedroudj et al., 2018). The word spatial domain according to the definition is working in the given space, in this case, the image. It entails working with the pixel values or working straight with the raw data. The spatial domain is the normal image space; another term used in this perspective is spatial derivative, where the image intensity values change if the position of the image is changed. However, researchers like (Kato et al., 2020) based their argument that even among various steganalysis, CNN is still promising. However, working on the spatial domain and spatial frequency to reduce the image gets high and subsequently disturbs the training process(Prajwalasimha et al., 2019). This shows the need for researchers to work on the spatial domain and frequency domain with the view to reducing the effect of disturbance in training.

The aforementioned technique involves utilizing cover-image properties, which can include payload or disturbance, t(Ye et al., 2017) to modify a grey-scale cover-image(X. Zeng et al., 2019). This results in a stego image with a small difference that is difficult for the human eye to detect(Ge et al., 2011). Spatial Domain Embedding is an important area of study for scholars to explore and develop different solutions. While this method may not be the most robust, its algorithms are simple and easy to analyze mathematically, making it a popular choice. The LSB method is a common technique for achieving this modification, either through arbitrary or serial changes to each pixel of the cover image(Yadollahpour& Naimi, 2009) or through LSB replacement or LSB matching(Muyco& Hernandez, 2019). Currently, feature-based steganalysis and machine learning are the most effective methods for noise residual computation, feature construction, and binary classification(Ye et al., 2017). In experiments comparing different cases using Support Vector Machines (SVM) and classifiers based on subjective Euclidean distance, improved results were achieved(You et al., 2021)

The study proposes a spatial-domain Rich Model (SRM) in the extraction process, The model is constructed into nine layers and segmented into a Three-stage CNN-based blind steganaylzer that accepts the raw image. Stacked Auto-Encoder forms the pre-training process layer-wise, which is the core clue of the SRM to capture various closer pixels among reliant. The second layer deals with pooling and subsampling to work with CNN to decrease dimensionality, truncation and SRM hardwired. The model equally looks into the structure of the filters and their diversity. The researcher took the image at 512 x 512 with 40 5 x 5 kernels used in the convolutional layers, and the maximum pooling layer is 4 x 4 downsampling as reported (Tan & Li, (2014). In another development, Cogranne and Fridrich, (2015) proposed steganographic plans that insert the mysterious message in a privately measured multivariate Gaussian cover image.

An all-inclusive contortion worldview that can be applied to inserting in a spatial area was planned by(Holub et al., 2014). The inserting contortion was registered as several relative changes of coefficients in a directional channel bank disintegration of the cover picture. The plan was performed in the spatial, JPEG, and side-informed JPEG domains. The proposed steganographic plans beat the present status of the craftsmanship steganographic methods(Holub et al., 2014). The paper affirms that confining the inserting changes to surfaces while staying away from clean edges significantly improves steganographic security(B. Li, Wei, et al., 2018). We analyzed and understood that the accuracy in deep learning image steganalysis needs improvement through an organized network plan, synthesis, learning approach, behaviours and mixing previous spatial domain methods into CNN structural design.

**Figure 1**: Taxonomy of Digital Image Transformation

In Figure 1, the taxonomy of digital image transformation classifies different techniques used for image processing and manipulation.Researchers and practitioners in computer vision and image processing can benefit from this resource, which shows a comprehensive framework for understanding and analyzing various techniques.More so, to improve safety most content-adaptive steganography necessitates embedding changes to control the texture areas where the statistics are solid to model in practice(Yan et al., 2021). Image steganalysis prompts the importance of improving information security, a better possibility in which the models like CNN images could reside to learn the features, with the view to improve image downsampling on CNN-based steganalysis using multiple steganography(Kato et al., 2020). The use of deep learning algorithms in steganalysis has made it possible tomerge and automate the two traditional stages(Gustavo & Reinel Tabares-SotoRaúl, 2019).

### 2.2 *Advanced Image Steganography*

Steganography creates a covert communication channel for hidden messages, implicit in different media types. The five types of steganographic methods are: spatial, transform, spread spectrum, statistical, and distortion(Holub et al., 2014)(Zhang et al., 2019). They cover extraction and training, usually for spatial-based space and change-based approaches(Signoroni et al., 2019)Therefore, Network Steganography is one of the maximum state-of-the-art and big steganography tactics having an adequate variety of approaches to transmit the name of the game data with none unique want of bodily cowl gadgets like Image, Video etc(B. Chen et al., 2020). Furthermore, Network steganography has attributes like un-detectability, bandwidth and robustness(Fridrich et al., 2007). In addition to retrieval of messages and haulier using TransSteg tools that are very effective in Network steganography, solutions are designed based on deep learning techniques of extraction and classification(Ye et al., 2017). The single-designed network is used to check the performance in enhancing stego-noise(B. Li, Li, et al., 2018). Components inserted into the architecture are to better the result. More so, the study observed the purview of steganographic algorithms such as S-UNIWARD, JUNIWARD and Hill-Cmd to demonstrate the strength of the model. However, the performance of present CNN-established approaches still could not overtake eccentric handcrafted feature-based approaches as reported(Erickson et al., 2017).

Most of the trendy adaptive steganography strategies first assign the distortion value. In which each pixel has a characteristic of the embedding information. All to reduce the predicted distortion value in the texture area. The size of entry images decreases as the texture increases. Handcrafted feature-based steganalysis and several CNN-based techniques in detection accuracy are featured in the review. A section with the most input to image feature extraction remains the way it is while low images or images with low probability are embedded(You et al., 2021). The result in Zhong & Chen, (2018) shows a 90% improvement when compared with other models like HUGO. The Gaussian dispersal with a typical derivative of 0.01 is used to prime the weight of matrices and use zero to start the bias vectors. Likewise, the Stochastic Gradient Descent (SGD) is used to enhance the network. The detection accuracies of trained show that the algorithm prototypes on diverse texture density images as 82.1% and 92.6% in WOW, and 81.4% and 90.0% in S-UNIWARD. The speed of the training model shows texture image is faster than the complex texture image, it takes 40 to 50 epochs of iteration to unite the network while the complex texture image takes about 70 to 80 epochs of iteration(J. Zeng et al., 2017). The GNCNN architecture combines the Qian-Net and Xu-Net models. Yet-Net and Chan-Net improve detection accuracy through data argumentation and single-channel modification. Shortcomings of Qian-Net include node non-convergence and lower performance than SRM. Xu-Net has low performance at the early stage and lacks proper CNN application(Xu et al., 2016). The proposed model will improve the performances of the GNCNN by applying hybridized activation functions (ELU, ReLU and LeakyReLU).

## 3.0 PROPOSED FRAMEWORK

Thus, a problem must exist before a model is designed, and then the type of dataset to used to improve the intricacy. Moreover, the complexity investigation of conventional machine learning models cannot be directly functional or openly extended to deep models(Gliner et al., 2021). The decision tree is built by splitting the input recursively and suggesting a local simple model for each region (Zhang et al., 2019). It is preferred for classification but is sometimes very useful for regression. Furthermore, the problem in the research work comprises both classification and regression in some stages. Likewise, the architecture of the deep learning process is a two-step approach at the top, and at the bottom, the unifying and automated approach is shown as a single-step approach in Fig 1:

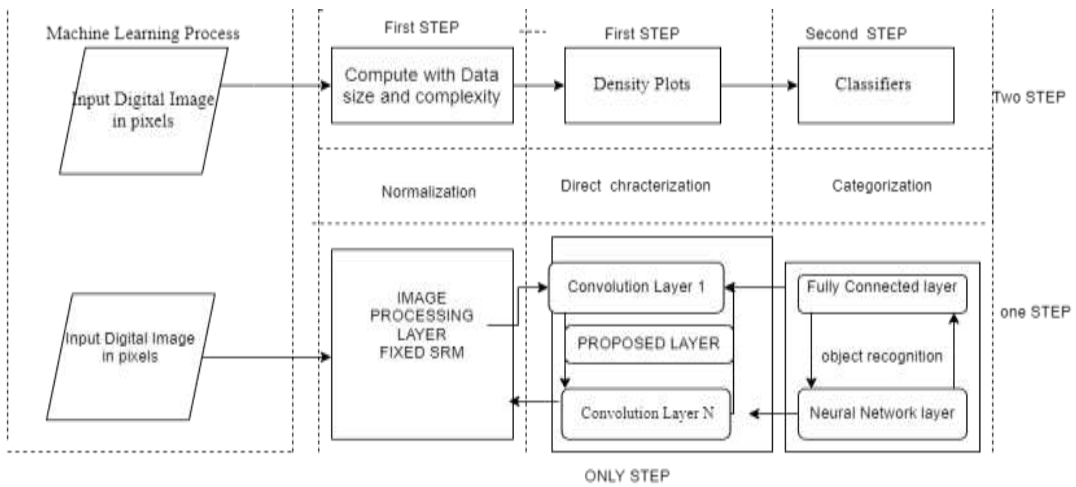


**Figure 2**: The Proposed Framework.

Figure 2: shows the proposed improved complexity of image steganalysis, the proposed techniques/ approaches to use deep learning-based feature extraction are Convolutional Neural Networks (CNNs) - CNNs are a popular choice for image classification and regression tasks(You et al., 2021). For the steganalysis aspect, it's proposed that CNNs are trained on a dataset of stego and cover images to extract features that are relevant for distinguishing between the two(Ke et al., 2019). The resulting feature vectors can then be fed into a classifier such as Support Vector Machines (SVMs) to make the final decision. In the pre-training model stage, the use of transfer knowledge becomes obvious, transfer learning involves reusing the knowledge gained

from training a deep neural network on a large dataset and applying it to a new task. In steganalysis, transfer learning can be used to fine-tune a pre-trained CNN on a dataset of stego and cover images(Shorten & Khoshgoftaar, 2019). In the research work, the acquired knowledge is implemented on the dataset and used in an environment called collaborator to run the pre-training test. How the model concept works in the hidden layer

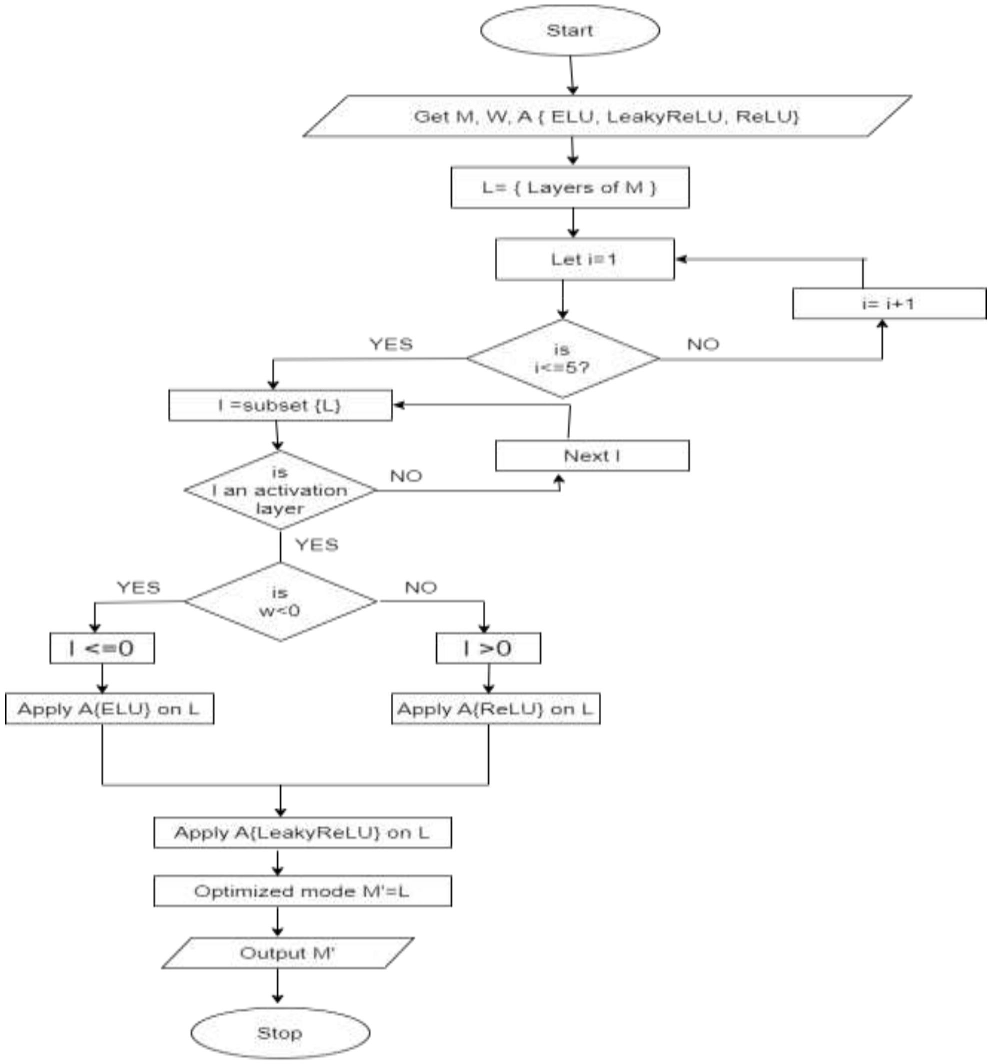


Fig 9: Flowchart of the Expanded Model Hidden Layer

**Figure 3: Flowchart of the Expended Model Hidden Layer**

### *3.4.1 Expended Model Hidden Layer*

The expanded model concept shown in Figure 3, displays the cycle of the hybridized activation function, that gives out a promising value of the digital image after running a test. First, all alfa value in the table above is set to default, while the filters keep on changing or increasing from 64 to 128 from layer 1 to layer 5 in the hidden layer. In deep learning, filter usually means the weights or parameters of a convolutional layer in a neural network. A convolutional layer uses a filter (also known as a kernel) to process the input data, creating a feature map that shows some features of the input. For instance, a filter can identify edges, shapes, colours, or patterns in a digital image. The filter size is the size of the filter, such as 3x3 or 5x5. The filter size can influence the effectiveness and efficiency of the convolutional layer, which is also part of the research outcome. Filters can be trained during training or designed for specific tasks. Moreso, seeing filters and feature maps can help to comprehend how a convolutional layer operates and what it learns. In addition, the dataset used is from a Kaggle competition known as Alaska 2image steganalysis. The training datasets are in the form of the cover image, JMIPOD, JUNIWARD, and USERD(Eid et al., 2022). As the model keeps on training the parameters change. Moreso, the parameters, are in the form of binary cross-entropy, which solves the loss function in the classification problem. The idea is to measure the difference between the predicted probability and the true labels of the dataset. Furthermore, the target is to improve the CNN network, deep learning uses different algorithms. The algorithm here is Adam optimizer, Adam “means” Adaptive Moment Estimation, which refers to adaptive learning rate and moment estimation capabilities. The algorithm is an extension of SGD which is used in updating the weights of a Neural Network. The choice for Adam is purely to improve the accuracy and speed of the NN. In addition, the model is sequential. Sigmoid and eLU are both activation functions that are employed in neural networks. They have some similarities and differences, such as

$$\boldsymbol{ELU}(\boldsymbol{x}) = \begin{cases} \quad\quad\quad\quad \boldsymbol{if}\ x \geq \mathbf{0} \\ \propto (\boldsymbol{e}^{x} - \mathbf{1})\ \boldsymbol{if}\ x < 0 \end{cases} \text{------------------- } \mathbf{(1)}$$

ELU activation Function

where α is a hyperparameter that regulates the value to which ELU approaches negative inputs.

The choice for the proposed stimulation function makes the average activations more centered around zero and lowers the bias shift. ELU has a smooth and continuous curve for all inputs, which prevents the abrupt change of ReLU and minimizes the gradient noise. ELU avoids the issue of vanishing gradients for negative inputs, which enhances learning speed and accuracy. Thus, making it an important component of the complexity of digital images. Similarly, ReLU stands for a rectified linear unit. It is defined as:

$$\boldsymbol{ReLU}(\boldsymbol{x}) = \mathbf{ma\,x}(\mathbf{0}, \boldsymbol{x})\text{---------------(2)}$$

ReLU activation Function

that it outputs zero for negative inputs and the input itself for non-negative inputs.

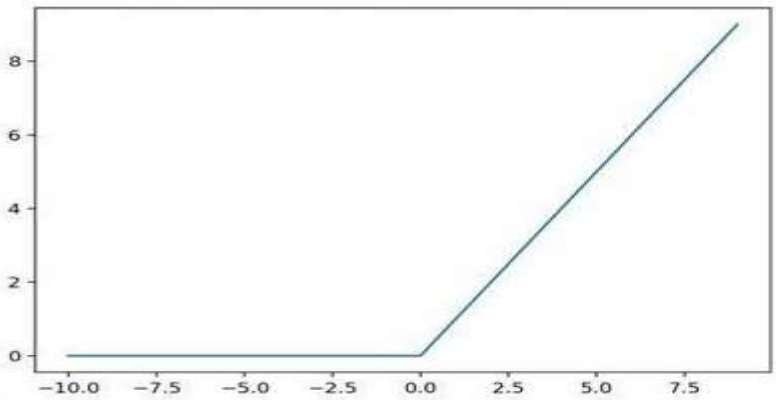


Figure 10: ReLU Activation function

Figure 10: displays the ReLU activation function working with only positive values. In addition, the research work takes into consideration the negative function as shown in Fig 10. Therefore, the ELU activation function serves as a flatter to revive the dying gradient. The activation function is a smooth function that can produce negative outputs, which can help avert the dying gradient problem of ReLU. Furthermore, leaky ReLU is shown in Fig 10.

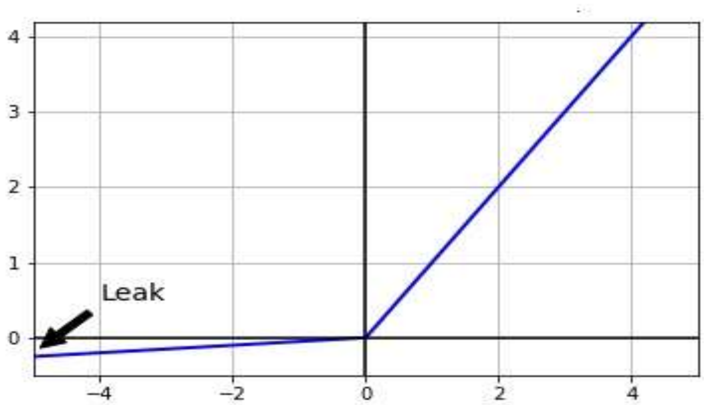


Fig 11: Leaky ReLU Activation Function

In Fig 11: leaky ReLU is displayed, The LeakyReLU activation isa variant of the function that has a small positive slope for negative inputs instead of zero. The slope coefficient α isusually a fixed hyperparameter LeakyReLU function is defined as

$$\boldsymbol{Leaky\ ReLU}(\boldsymbol{x}) = \begin{cases} x\ if\ \ x \geq \mathbf{0} \\ \boldsymbol{a}x\ \ if\ x < 0 \end{cases}\text{-----------------------(3)}$$

LeakyReLU function with α=0.1:

Equation 3 is a leaky ReLU function showing how leaky ReLU is defined, However, the added activation function can be slower to compute than ReLU because of the exponential operation. However, ReLU can encounter the dying gradient problem, thus, some neurons can become inactive and stop learning if the inputs are always negative. Therefore, the need to add another activation function in the hybridization became necessary. The default values of all the activation functions were set up from 0 to infinity for ReLU, while leaky ReLU is set up from 0.01 this represents the alfa values. Exponential Linear Unit (ELU) is hybridized with the two activation functions to improve outcomes. ELU was introduced in 2015 by Djork-Arne Clevert, Thomas Unterthiner, and Sepp Hoctreiter. ELU takes care of negative inputs in the process.

Typically, Traditional learning processes have limitations, and deep learning methodologies come with specific hyperparameters that must be fine-tuned for optimal performance. Hyperparameters offer several advantages, including reducing manual effort, enhancing performance, and increasing efficiency(Lavesson and Davidsson, 2006), (Mantovani et al., 2015), The learning rate hyperparameter governs the model's pace of learning, while the epoch refers to the total number of iterations over the training dataset(Hu et al., 2021). The dropout rate enhances the model's fitting capacity and prevents overfitting, while the embedding layer converts words into fixed-length vectors. The max-depth parameter governs the maximum depth of the tree, and the n-estimator parameter generally augments model performance(Probst et al., 2018) (Van et al., 2018).

The enhancement is made in the max pooling, by proposing additional layers and changing the max pooling from 32 to 64. Then filters are increased to 96 up to 128 in the hidden layer to optimize the output. Furthermore, three hidden layers are added to the model to make it more robust excluding the input and the output layer. More so, an additional epoch is considered, rising to 30 to 50 epochs to optimize our model better.

Similarly, a proposed kernel increase existed, a kernel is a filter that is used to extract the features from the images. The complexity and performance of a CNN depend on the kernel size, which is a hyperparameter. A larger kernel size implies a larger number of parameters and a larger receptive field of the network. However, this also implies a higher computational cost and a higher possibility of overfitting. Thus, a trade-off between accuracy and efficiency is a major factor considered in selecting the kernel size. Cross-validation or grid search is used to determine the best kernel size for the model. Furthermore, in this research work, the kernel size is carefully selected to ensure better training and validation. Consequently, the kernel is a matrix (5,5) that moves over the input dataset, performs the procedure with the sub-region of the input data, and gets the output. The reason for the choice is to ensure effectiveness and efficiency are achieved. Accordingly, the kernel moves on the input data as it moves with the stride value. The stride moves left and right, top to bottom with pixel column change on horizontal movement and one vertical movement too. In addition, the stride changes are a result of the idea to downsample the input by aggregating neighbouring pixels or features, to make the computation more efficient and reduce overfitting. Thereby, to make the inputs in the network more consistent, a batch normalization layer is introduced. Batch normalization is a technique that normalizes either based on the outputs of the previous layer or the original inputs. So that transforms inputs to a mean of 0, and a standard deviation of 1 is achieved. This helps to speed up the training process and reduce overfitting. The batch normalization is used before the activation functions.

Another improvement to the hybridized model is in the max pooling, max pooling is a type of pooling technique that involves taking the supreme value within each rectangular area of the feature map. Furthermore, in the model training, after the convolutional layer in CNNs, max pooling was used (3,3). The highest value in each pixel window is shrinking, thereby making the features good and lowering the computational cost. In addition, the pooling was specified giving transparency over the input, while the stride was moved each time by the window. As a result, the output of a max pooling layer has fewer stimulations compared to the input and highlights the most significant feature within each region. The impression exploited the efficiency and the ability to decrease the spatial dimensionality of the feature maps. Thus, making the CNN architecture more resistant to small disparities in the input digital image.

Dropout is also an important element used in this study. The choice is connected with the likely overfitting that is expected when training a model. The technique reduced overfitting in NN. This is achieved by arbitrarily ignoring some nodes or units in a layer in the training session, thus, making the network robust. For instance, the dropout rate p is the chance of a neuron being dropped out of the network in each training step. The neurons that are dropped out are randomly selected in every training step. So, a neuron that is dropped out in one training step can still be active in the next one with a probability of $(1-p)$. Note the dropout is used only in the training stage. Similarly, Flattening is a procedure that is used to convert multi-dimensional arrays into a 1-D array. Thus, feeding the 1-D array to the classification model. Equally, the model made use of Con2D, the layer applies convolution operation on the input data using a kernel size and shape. Accordingly, the fully connected layer which is also known as the dense layer is used in the proposed research work. Since dense layers exist in various libraries including TensorFlow or Keras. For the reason that the dense layer varies in the number of neurons, which affects the output of the layer. Subsequently, it's linked to all input layers to all neurons in the current layer then the activation functions too are affected. Therefore, it performs a non-linear transformation to the output.

### *3.5 Dataset*

In the proposed research, two datasets have been used - the training dataset and the validation or testing dataset. A dataset in deep learning is a set of samples that are used to teach, evaluate, or measure a deep learning model. It can be divided into different parts for different objectives, such as training, validation, and test parts. The research focuses on image analysis and cybersecurity, where the model can detect the gradations of image steganalysis and manipulation. Out of the total dataset, 80% is allocated for training purposes, and the remaining 20% is reserved for testing. The datasets are in the form of cover, juniward, uerd, mipod, and test datasets over iteration. By utilizing these datasets, algorithms can be developed to improve the detection of manipulated images, which can help prevent various forms of digital image tampering or manipulation. It is crucial to use these datasets ethically and responsibly to make a positive contribution to the fields of cybersecurity and digital forensics.

Likewise, the proposed model used splitting, scaling, and normalization, since we are using digital images. The steps are as follows, First, we split the data into training and testing. This is to guarantee there is no overfitting and measure the general performance of the proposed model. Likewise, for proper optimization, normalizing the input values is applied. Secondly, the validation or testing, the outcome is suitable for the variety of activation functions used while optimizing the gradient descent process. In addition, to prevent data leakage in the proposed model, features are transformed to a similar scale after dividing the data, using statistics from the training. Another important component of research work is the metrics for measuring the effectiveness of our model. The metrics include accuracy, precision, F1 score, and Recall.

In highlighting the importance ofhow the nodes received datasets, figure 12 shows that the map input to output takes negative input into account in the proposed model algorithm. The expression is a piecewise function. A piecewise function consists of several sub-functions, each of which is valid for a specific part of the domain. The choice of an open-source library is adopted for this research work; therefore, the TensorFlow library is used. TensorFlow released in 2015 is a multidimensional array of numbers scalar and matrix-vector software for building and deploying ML and deep learning models. The important includes computation of a graph of data flow. Which consists of a node, edge, and data, The node represents mathematical operation, the edge represents data that is communicated from one node to another, and data in tensor flow is represented as a tensor which is a multidimensional array.

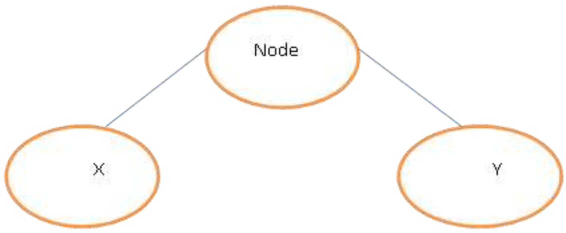


**Fig 12:** Typical Example of a data flow

An example of data flow is shown in Figure 12: Data flow is important in the hidden layer of the activation function because it determines how the information from the input layer is transformed and propagated to the output layer. Thus, the distribution and handling of different operations or nodes in a neural network is what is referred to as data flow(Zhou et al., 2020). The node in the DFD is a data store, an external entity, or a process. Data flows that enter or exit a node have labels X and Y(Yang et al., 2017). For example, X is the input data name for the process node, and Y is the output data name for the same node. X and Y can also be variables that represent the data values that are manipulated or stored by the nodes. Therefore, after the dataflow other tools need to be discussed such as Keras a high-level programming tool kit that makes it easy to build many different types of NN with only a few lines of code is our choice. Keras is a popular wrapper for TF(Ravi & Alazab, 2021). The choice of environment is the Google collab. In general, the dataset is Alaska2 image steganalysis BossBase 1.0 from Kaggle for competition(Ravi & Alazab, 2021).

### *3.6 Hybridized Algorithm*

To optimize the activation function, the hybridization of the model was deemed necessary. The following table presents alternative algorithmic approaches, highlighting their respective benefits, limitations, and motivation functions. These methods serve to demonstrate additional methodologies that can aid in achieving the model's objectives.

Table 1:

| **Algorithms and Year** | **Benefits** | **Shortcomings** | **Motivation Function** |
|---|---|---|---|
| Yedroudj-Net 2018 | The model uses pre-processing about 30-filter bank, truncation activation, and five convolutional layers with Batch Normalization linked using a Scale Layer. The model is better than Ye-Net and Xu-Net | Complexity of the mode, convergence time increases | ReLU |
| Zhong-Net 2019 | The model uses the Image texture feature, a gray-level co-occurrence matrix. Several data sets are divided into subsets of different intricacy based on the index steganalysis are developed and trained in the MER-Most Effective Region and initial ideas. | Network generalization capability on the adaptive algorithm is weak | ReLU |
| WISER-Net 2019 | Steganalysis structure for JPEG color images. Channel-wise convolution was introducedto the network which makes performance superior | The modelis very complex with mismatching | ReLU |
| You-Net (SiaStegNet) 2020 | Siamese, CNN-based architecture, which consists of two symmetrical subnets with shared parameters, and contains three phases: preprocessing, feature | Methodology yet to solve arbitrary-size image steganalysis, | ReLU and TanH |

| | | | |
|---|---|---|---|
| | extraction, and fusion/classification. Unfixed network layers. Datasets are heterogeneous | Mismatch in the statistical distribution caused by resizing of images. | |
| CALPA-NET 2020 | The hybrid criterion makes use of three-stage that includes, training, pruning, and fine-tuning to protect a set of inherited learned important weights<br>The convolutional layer is in a data-driven manner.<br>The model shows adaptivity, transferability, and scalability. | CALPA- SRNet is slightly worse than the original SRNet when JPEG stego images are with Q F 95. This indicates that when used to detect 0.4 bpp<br>Redundant parameters make it slower. | Truncated Linear Unit (TLU) |

Table 1: above shows different models from 2018 and 2020. Showing some activation functions ReLU and TLU, some ReLU and Tanh, and some ReLU only. Therefore, in this research study, a clear distinct method of hybridizing different activation functions is presented. Wherethe hybridized algorithm presents the step-by-step of the new proposed model. Thus, the CNN architecture for the model was designed by hybridizing different activation functions, with varying parameters, kernel values, max-pooling layers, and dropout layers.

In the research work, the hybridized algorithm model has shown promising results.

The hybridized model is shown as:

```
Function HybridizeLUActivationFunctionInCNNForSteganalysis
Input:
        Let M be the model
        Let W be the input weighted vectors to the model
        Let A {ELU, LeakyReLU, ReLU} be the set of activation functions
Output:
        A modified model M'
Algorithm:
        Let L be the set of layers of M
        For i = 1 to 5
            For each l in L
                If l is an activation layer, Then
                    If W < 0 Then
                        Select l' from A {ELU}
                    Else If W > 0 Then
                        Select l' from A {ReLU}
                    End If
                End If
                If l' is still <= 0 Then
                    Select l' from A {LeakyReLU}
                End If
            End For each
            Replace l with l' in L
        End For
    Create a new improved model M' from L
End
```

## 4.0 RESULTS

4.1 In this section the research presents results and analysis showing the three algorithms used to improve complexity in image steganalysis. The result improves the problem of vanishing gradient and dying neurons. The following parameters are used in our training and testing; they are;

The parameters setting during the training phase and testing phase of the proposed approach

Table 1: Hyper-parameter values

| **Hyper-parameter** | **values** |
|---|---|
| Learning rate | 0.0001 |
| Maximum Epoch | 50 |
| Dropout rate | 0.2 |
| Batch- size | 32 |

| | |
|---|---|
| Validation sample | 0.20 |
| Training sample | 0.80 |
| Kernel size | (5,5) |
| Max pooling | 5 by 5 |

### *4.2 EXPERIMENTAL SETUP AND IMPLEMENTATION DETAILS*

### *4.3 Presentation and Discussion of Results*

Table 2: Performance Accuracy of the Proposed Activation Function

| activation function | Accuracy (%) |
|---|---|
| ELU | 75 |
| **RELU** | 74.9 |
| **Leaky RELU** | 75.25 |
| **Hybridized activation function.** | 76.1 |

Table 2 shows that the proposed hybridized activation function hasa high accuracy of 81%, which indicates that it performed better than the existing ones. Thus, making more correct predictions, however, it's important to note that accuracy may not be the only relevant metric in all scenarios, especially when dealing with imbalanced datasets or specific applications where other metrics like precision, recall, or F1 score may be more informative.

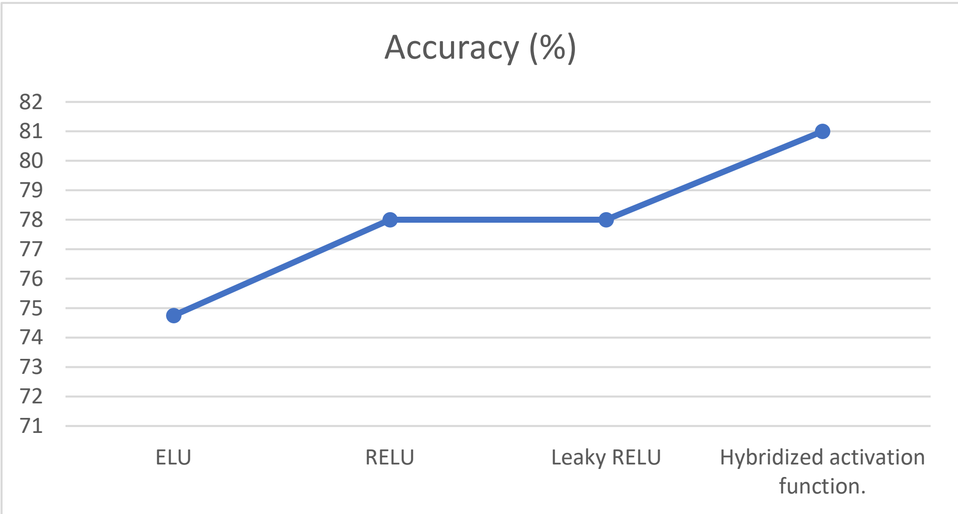


Figure 3: Graph Showing Accuracy

In Figure 3; Accurate activation functions are indeed crucial for correct predictions. Thereby the proposed hybridized activation functions performed better against the other function with 81% accuracy. As such, the proposed activation function outshined Elu, Relu and leakyRelu.

Table 3: Precision table showing performance results.

| **Activation function** | **Precision (%)** |
|---|---|
| ELU | 74.75 |
| RELU | 78 |
| Leaky RELU | 78 |
| Hybridized activation function. | 81 |

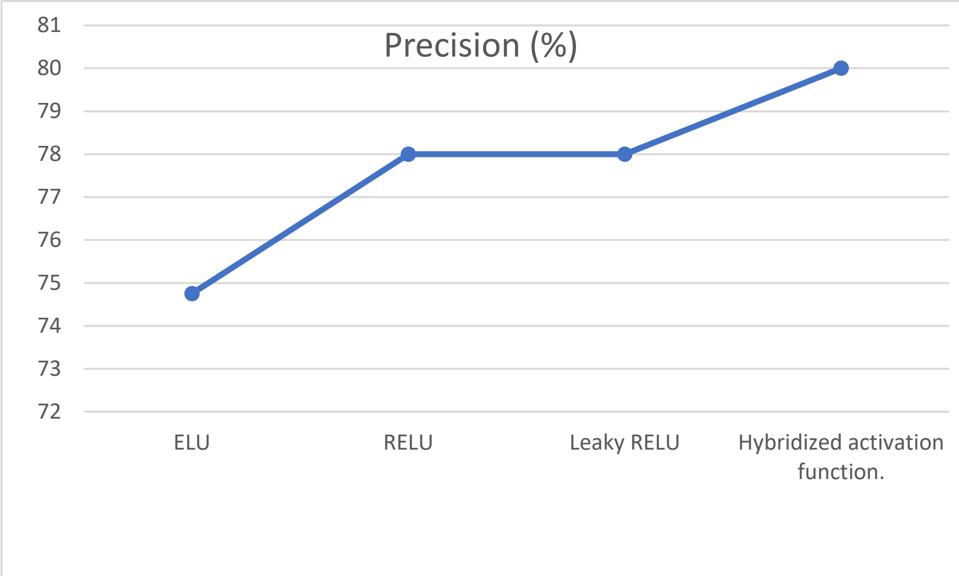


Figure 4: Graph showing Precision

The graph represents results from CNN + HAF model training and validation, achieving over 80% precision, surpassing other models.

Table 4: F1-Score table showing f1-score performance

| **MODELs** | **F1-Score (%)** |
|---|---|
| ELU | 85.55 |
| RELU | 85.55 |
| Leaky RELU | 87.64 |
| Hybridized activation function. | 89.90 |

Table 4, shows that the hybrid model obtained 89.90% which is better than the existing models.

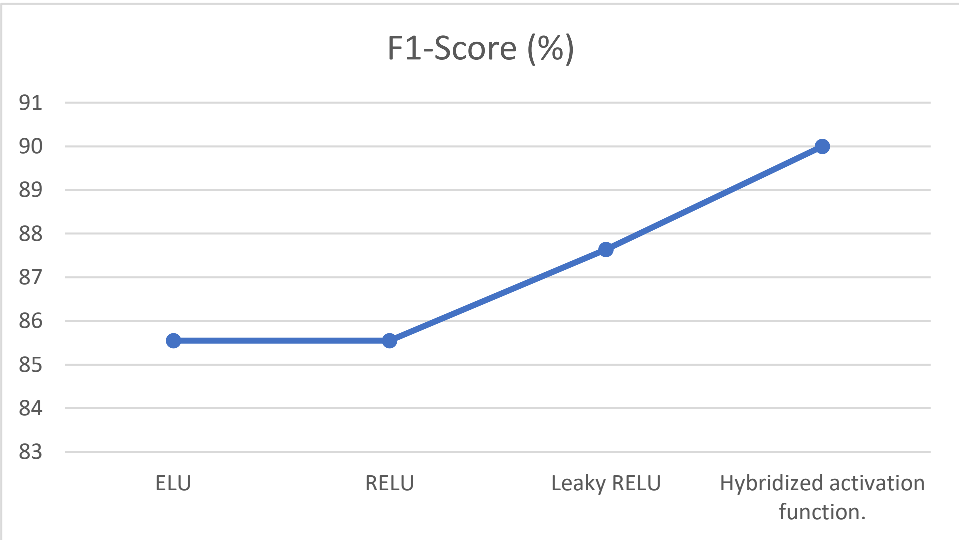


Figure 5; Graph showing F1-score performance

Table 5:

| **Activation function** | **Recall (%)** |
|---|---|
| ELU | 100 |
| RELU | 100 |
| Leaky RELU | 100 |
| Hybridized activation function. | 100 |

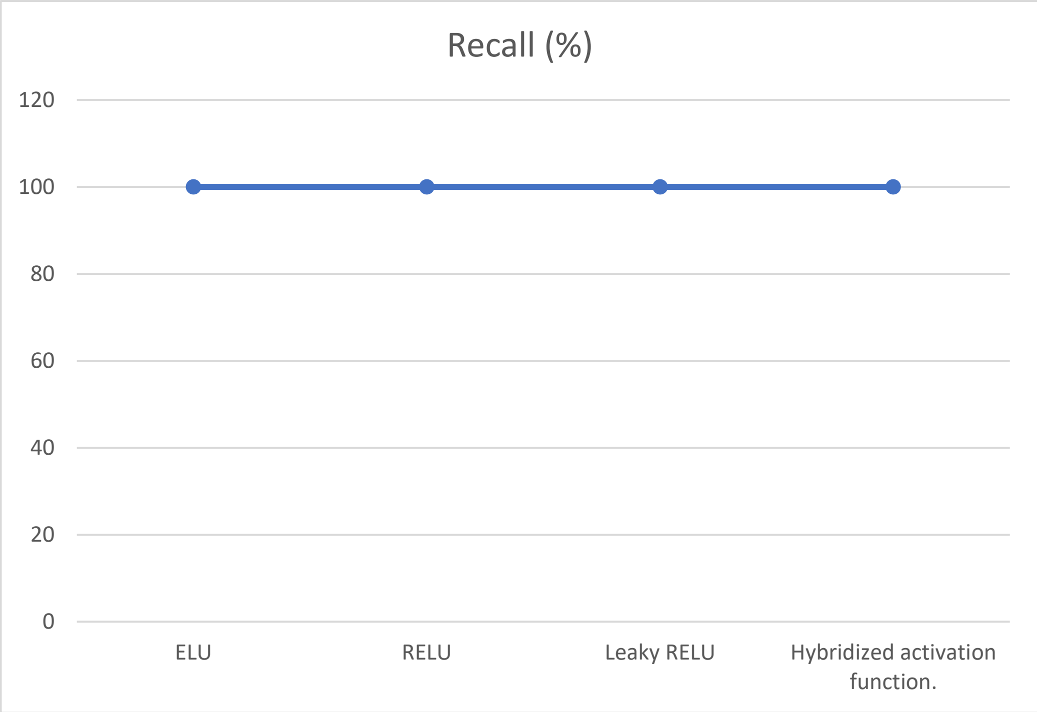


Figure 6: Graph showing recall performance against all models.

In Figure 6, they the show same recall across the proposed model and the existing models.

Table 6: Comparative Analysis of the proposed model against the existing models

| Reference | MODELs | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|---|
| **Existing work** | ELU | 74.11 | 74.75 | 100 | 85.55 |
| | RELU | 78.00 | 78.00 | 100 | 85.55 |
| | Leaky RELU | 78.00 | 78.00 | 100 | 87.64 |
| **Proposed** | Hybridized activation function. | 81.00 | 80.10 | 100 | 89.51 |

Based on Table 6, it can be observed that the higher the activation performance in percentage, the better the overall performance of the function. Conversely, a lower activation function performance in percentage leads to a weaker performance of the function. The proposed hybridized activation function delivered the best performance in terms of Accuracy, Precision, Recall, and F1-score. As shown in Table 6, the hybridized activation function achieved 76%, 76%, and 86.36% for Accuracy, Precision, and F1-score respectively, when compared to YOU-NET and GNCNNin terms of accuracy at 50 epochs. accuracy. In summary, both Table 6 and Figure5 demonstrate that the hybridized function is accurate, precise, efficient, and optimized when compared to the existing activation functions. Similarly, Figure5 indicates that the recall intersects at all activation functions because they have instances at a particular class. It is worth noting that there is a meet point of a function in our matrix of choice.

Activation functions play a crucial role in enabling neural networks to model non-linear relationships, extract features, make classification decisions, and facilitate the training process.

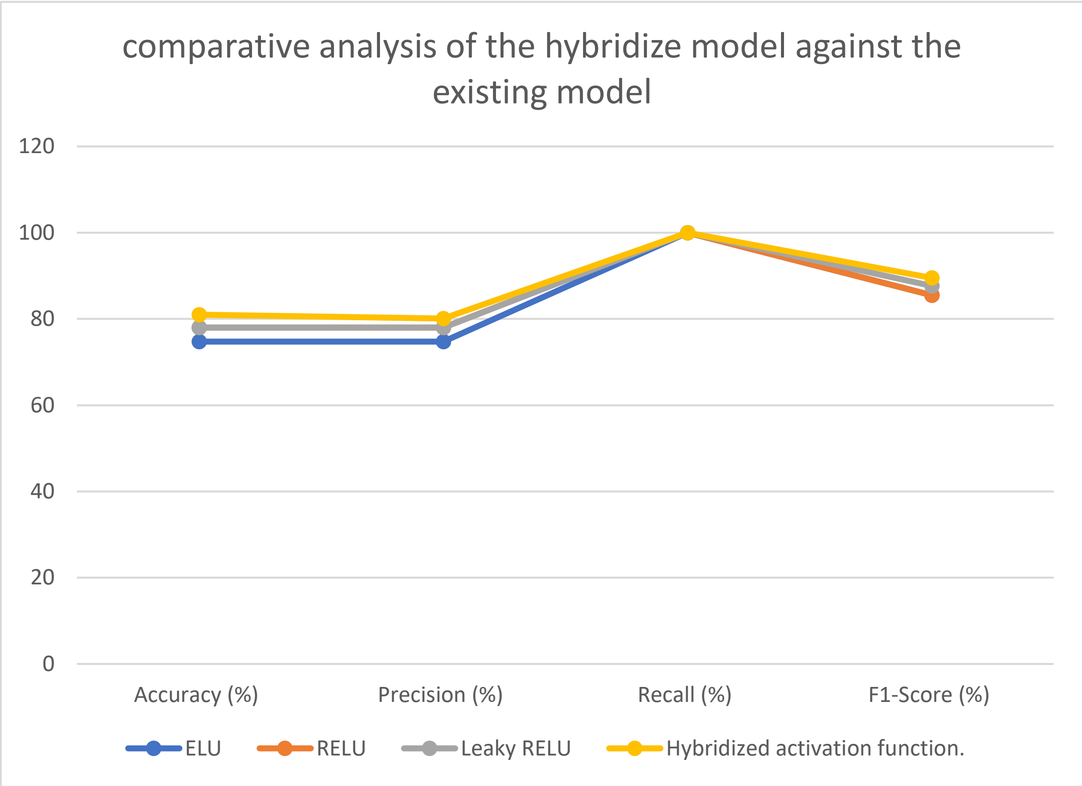


Fig 7:Graph comparative analysis of the hybridized model against the existing model

## 5.0 Summary

The research has shown that the proposed CNNHAF performs better in terms of accuracy, achieving 81%, compared to existing models. This indicates that combining functions has resulted in an improved model that can effectively handle negative slopes in an image dataset and smoothly address the issue of dead neurons in a CNN model. Additionally, the efficiency of the models is presented graphically across five metrics. This is a crucial aspect of improving deep learning in today's cyber world.

### *5.1 Conclusion*

Thus, Conventional activation functions like ELU, ReLU, and Leaky ReLU have been widely used, but they come with certain limitations, including computational cost and the issue of dead neurons. To address these limitations, an innovative approach has been introduced, presenting a hybrid activation function that leverages the power of a convolutional neural network (CNN) to streamline feature extraction and classification processes. In addition, this innovative activation function surpasses the performance of existing ones, achieving an impressive accuracy improvement of 81%. This advancement in activation function design not only enhances the effectiveness of steganalysis but also the accuracy, precision, recall, and f1-score metrics. However, it also has a drawback in resources and performance speed due to the exponential in linear units but also holds promise for broader applications in deep learning and neural network-based tasks. The model's noise signal is not investigated. Likewise, hyper parameter can be fine tune more to achieve a better result.

### *5.2 Recommendation*

The research recommends a thorough investigation into several critical factors to improve the performance of the combined activation function. These factors include early stopping, model biases and weights, gradient size, and signal noise. A comprehensive understanding of these factors is vital to attain optimal performance of the innovative combined activation function. Therefore, it is advised to conduct a detailed analysis of these factors, which could significantly enhance the overall efficiency of the model.

**REFRENCES**

Aggarwal, C. C. (2018). Neural Networks and Deep Learning. In Neural Networks and Deep Learning. https://doi.org/10.1007/978-3-319-94463-0

Atta, R., & Ghanbari, M. (2021). A high payload data hiding scheme based on dual tree complex wavelet transform. Optik, 226, 165786. https://doi.org/10.1016/j.ijleo.2020.165786

Chen, B., Luo, W., Zheng, P., & Huang, J. (2020). Universal stego post-processing for enhancing image steganography. Journal of Information Security and Applications, 55. https://doi.org/10.1016/j.jisa.2020.102664

Chen, M., Sedighi, V., Boroumand, M., & Fridrich, J. (2017). JPEG-phase-aware convolutional neural network for steganalysis of JPEG images. IH and MMSec 2017 - Proceedings of the 2017 ACM Workshop on Information Hiding and Multimedia Security, 75–84. https://doi.org/10.1145/3082031.3083248

Chutani, S., & Goyal, A. (2019). A review of forensic approaches to digital image Steganalysis. Multimedia Tools and Applications, 78(13), 18169–18204. https://doi.org/10.1007/s11042-019-7217-0

Cogranne, R., & Fridrich, J. (2015). Modeling and Extending the Ensemble Classifier for Steganalysis of Digital Images Using Hypothesis Testing Theory. IEEE Transactions on Information Forensics and Security, 10(12), 2627–2642. https://doi.org/10.1109/TIFS.2015.2470220

Eid, W. M., Alotaibi, S. S., Alqahtani, H. M., & Saleh, S. Q. (2022). Digital Image Steganalysis: Current Methodologies and Future Challenges. IEEE Access, 4. https://doi.org/10.1109/ACCESS.2022.3202905

Erickson, B. J., Korfiatis, P., Akkus, Z., Kline, T., & Philbrick, K. (2017). Toolkits and Libraries for Deep Learning. J Digital Imaging, 30, 400–405. https://doi.org/10.1007/s10278-017-9965-6

Fridrich, J., & Kodovsky, J. (2012). Rich models for steganalysis of digital images. IEEE Transactions on Information Forensics and Security, 7(3), 868–882. https://doi.org/10.1109/TIFS.2012.2190402

Fridrich, J., Pevný, T., & Kodovský, J. (2007). Statistically undetectable JPEG steganography: Dead ends challenges, and opportunities. In J. Dittmann & J. Fridrich (Eds.), MM and Sec'07 - Proceedings of the Multimedia and Security Workshop 2007 (pp. 3–14). Dallas. https://doi.org/10.1145/1288869.1288872

Ge, H., Huang, M., & Wang, Q. (2011). Steganography and steganalysis based on digital image. Proceedings - 4th International Congress on Image and Signal Processing, CISP 2011, 1, 252–255. https://doi.org/10.1109/CISP.2011.6099953

Gliner, J. A., Morgan, G. A., Leech, N. L., Gliner, J. A., & Morgan, G. A. (2021). Measurement Reliability and Validity. Research Methods in Applied Settings, 319–338. https://doi.org/10.4324/9781410605337-29

Gustavo, I., & Reinel Tabares-SotoRaúl, R. P. (2019). Deep Learning Applied to Steganalysis of Digital Images : A Systematic Review. IEEE Access, 7. https://doi.org/10.1109/ACCESS.2019.2918086 Deep

Holub, V., Fridrich, J., & Denemark, T. (2014). Universal distortion function for steganography in an arbitrary domain. Eurasip Journal on Information Security, 2014, 1–13. https://doi.org/10.1186/1687-417X-2014-1

Hu, X., Chu, L., Pei, J., Liu, W., & Bian, J. (2021). Model complexity of deep learning: a survey. Knowledge and Information Systems, 63(10), 2585–2619. https://doi.org/10.1007/s10115-021-01605-0

Hussain, I., Zeng, J., Xinhong, & Tan, S. (2020). A survey on deep convolutional neural networks for image steganography 1. Hussain I, Zeng J, Xinhong, Tan S. A survey on deep convolutional neural networks for image steganography and steganalysis. KSII Trans Internet Inf Syst. 2020;14(3):1228–48. and ste. KSII Transactions on Internet and Information Systems, 14(3), 1228–1248. https://doi.org/10.3837/tiis.2020.03.017

Kato, H., Member, G. S., & Osuge, K. (2020). A Preprocessing by Using Multiple Steganography for Intentional Image Downsampling on CNN-Based Steganalysis. IEEE Access, 8. https://doi.org/10.1109/ACCESS.2020.3033814

Ke, Q., Ming, L. D., & Daxing, Z. (2019). Image steganalysis via multi-column convolutional neural network. International Conference on Signal Processing Proceedings, ICSP, 2018-Augus, 550–553. https://doi.org/10.1109/ICSP.2018.8652324

Li, B., Li, Z., Zhou, S., Tan, S., & Zhang, X. (2018). New Steganalytic Features for Spatial Image Steganography Based on Derivative Filters and Threshold LBP Operator. IEEE Transactions on Information Forensics and Security, 13(5), 1242–1257. https://doi.org/10.1109/TIFS.2017.2780805

Li, B., Wei, W., Ferreira, A., & Tan, S. (2018). ReST-Net: Diverse Activation Modules and Parallel Subnets-Based CNN for Spatial Image Steganalysis. IEEE Signal Processing Letters, 25(5), 650–654. https://doi.org/10.1109/LSP.2018.2816569

Li, H., Brandt, J., Lin, Z., Shen, X., & Hua, G. (2016). A multi-level contextual model for person recognition in photo albums. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem, 1297–1305. https://doi.org/10.1109/CVPR.2016.145

Muyco, S. D., & Hernandez, A. A. (2019). A modified hash based least significant bits algorithm for steganography. ACM International Conference Proceeding Series, 215–220. https://doi.org/10.1145/3335484.3335514

Prajwalasimha, S. N., Chethan, S. S., & Mohan, C. S. (2019). Performance analysis of DCT and successive division based digital image watermarking scheme. Indonesian Journal of Electrical Engineering and Computer Science, 15(2), 750–757. https://doi.org/10.11591/ijeecs.v15.i2.pp750-757

Qian, Y., Dong, J., Wang, W., & Tan, T. (2017). Feature learning for steganalysis using convolutional neural networks. Springer Science. https://doi.org//doi.org/10.1007/s11042-017-5326-1

Ranjan, R., & Forensics, C. (2016). JPEG IMAGE STEGANALYSIS USING MACHINE. IJCSI International Journal of Computer Science and Information Security, 14(2), 96–99.

Ravi, V., & Alazab, M. (2021). Deep Learning for Cyber Security Applications : A Comprehensive Survey. ResearchGate, October. https://doi.org/10.36227/techrxiv.16748161

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1). https://doi.org/10.1186/s40537-019-0197-0

Signoroni, A., Savardi, M., Baronio, A., & Benini, S. (2019). Deep learning meets hyperspectral image analysis: A multidisciplinary review. In Journal of Imaging (Vol. 5, Issue 5). https://doi.org/10.3390/jimaging5050052

Tan, S., & Li, B. (2014). Stacked convolutional auto-encoders for steganalysis of digital images. 2014 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA 2014, 2012lym 0117. https://doi.org/10.1109/APSIPA.2014.7041565

Wang, Z., Chen, M., Yang, Y., Lei, M., & Dong, Z. (2020). Joint multi-domain feature learning for image steganalysis based on CNN. Eurasip Journal on Image and Video Processing, 2020(1). https://doi.org/10.1186/s13640-020-00513-7

Xu, G., Wu, H. Z., & Shi, Y. Q. (2016). Ensemble of CNNs for steganalysis: An empirical study. IH and MMSec 2016 - Proceedings of the 2016 ACM Information Hiding and Multimedia Security Workshop, 103–107. https://doi.org/10.1145/2909827.2930798

Yadollahpour, A., & Naimi, H. M. (2009). Attack on LSB steganography in color and grayscale images using autocorrelation coefficients. European Journal of Scientific Research, 31(2), 172–183.

Yan, X., Lu, Y., & Wang, S. (2021). Image Steganalysis via Diverse Filters and Squeeze-and-Excitation Convolutional Neural Network.

Yang, J., Shi, Y.-Q., Wong, E. K., & Kang, X. (2017). JPEG Steganalysis Based on DenseNet. 1–7. http://arxiv.org/abs/1711.09335

Ye, J., Ni, J., & Yi, Y. (2017). Deep Learning Hierarchical Representations for Image Steganalysis. IEEE Transactions on Information Forensics and Security, 12(11), 2545–2557. https://doi.org/10.1109/TIFS.2017.2710946

Yedroudj, M., Comby, F., & Chaumont, M. (2018). Yedroudj-net: An efficient cnn for spatial steganalysis. IEEE, 1, 2092–2096.

You, W., Zhang, H., & Zhao, X. (2021). A Siamese CNN for Image Steganalysis. IEEE Transactions on Information Forensics and Security, 16, 291–306. https://doi.org/10.1109/TIFS.2020.3013204

Zeng, J., Member, S., Tan, S., Member, S., Li, B., & Member, S. (2017). Large-scale JPEG image steganalysis using hybrid. IEEE Transactions on Information Forensics and Security, 6013(c), 1–15. https://doi.org/10.1109/TIFS.2017.2779446

Zeng, X., Feng, G., & Zhang, X. (2019). Detection of double JPEG compression using modified DenseNet model. Multimedia Tools and Applications, 78(7), 8183–8196. https://doi.org/10.1007/s11042-018-6737-3

Zhang, C., Yu, J. J. Q., & Liu, Y. (2019). Spatial-Temporal Graph Attention Networks: A Deep Learning Approach for Traffic Forecasting. IEEE Access, 7, 166246–166256. https://doi.org/10.1109/ACCESS.2019.2953888

Zhou, X., Hong, W., Weng, S., Chen, T. S., & Chen, J. (2020). Reversible and recoverable authentication method for demosaiced images using adaptive coding technique. Journal of Information Security and Applications, 55(November), 102629. https://doi.org/10.1016/j.jisa.2020.102629