Taylor & Francis
Taylor & Francis Group

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

A security and privacy preserving approach based on social IoT and classification using DenseNet convolutional neural network

C. Maniveena & R. Kalaiselvi

To cite this article: C. Maniveena & R. Kalaiselvi (2024) A security and privacy preserving approach based on social IoT and classification using DenseNet convolutional neural network, Automatika, 65:1, 333-342, DOI: 10.1080/00051144.2023.2296788

To link to this article: https://doi.org/10.1080/00051144.2023.2296788

Published online: 08 Jan 2024.

AUTOMATIKA
2024, VOL. 65, NO. 1, 333–342
https://doi.org/10.1080/00051144.2023.2296788

Taylor & Francis
Taylor & Francis Group

A security and privacy preserving approach based on social IoT and classification using DenseNet convolutional neural network

C. Maniveena[a] and R. Kalaiselvi[b]

[a]Department of Computer Science and Engineering, Noorul Islam Centre for Higher Education, Thuckalay, India; [b]Department of Computer Science and Engineering, RMK College of Engineering and Technology, Puduvoyal, India

ABSTRACT

This method is able to synthesize fine-detailed images by the use of a global attention that gives more attention to the words in the textual descriptions. Also we have the deep attention multimodal similarity model (DAMSM) that calculates the matching loss in the generator. Though this work produced images of high quality, there was some loss while training the system and it takes enough time for training. Although there has been little study on applying character-level Dense Net algorithms for text classification tasks; the Dense Net structures we suggested in this paper have shown outstanding performance in image classification tasks. Extensive testing has revealed that they perform better when it comes to their ability to withstand interruption and that they can influence exerted many organizations implementing information usage and language information on the specifications of user privacy protection, framework implies, and regulatory requirements.

ARTICLE HISTORY
Received 5 September 2023
Accepted 10 December 2023

KEYWORDS
Security; privacy preservation; text classification; DenseNet; social IoT

1. Introduction

A common issue in many applications is text conversion. Many approaches, including stack GAN, were used to synthesize images from the provided textual descriptions. Even though each technology was capable of creating high-resolution photographs, it also had its own drawbacks. The attention GAN, whose extensive work is being mirrored in this approach, is another technique to synthesize images from natural literary devices. By using a global attention strategy that places a greater emphasis on the words in the textual descriptions, this technique can synthesis finely detailed visuals.

Additionally, the deep awareness multimodal similarity model (DAMSM), which determines the matching loss in the generator, is available. Despite the fact that this study produced high-quality photos, some losses occurred while the system was being trained, and training takes a long time. In order to decrease loss and training time and produce images with more unique features, the DenseNet architecture is proposed in this study. The inception architecture was employed in the prior model, but because of the wider distance between the output layer and input layer, it lost effectiveness during training and took a long time. Each layer in DenseNet is interconnected, allowing the input from one layer to be passed on to another. This method was able to speed up result retrieval while reducing loss by 10%. Additionally, the DAMSM determines the matching loss in the generator. We used the CUB dataset for this, which had 12,000 texts from 200 different birds with 10 captions for each sentence. We separated the data into training and testing sets using random distribution division, with 49.2% and 50.8% of the total data, respectively.

The use of deep learning techniques in recent years has greatly aided natural language processing (NLP). Particularly, text categorization tasks have seen substantial performance gains. Long short-term memory (LSTM) [6] frameworks have traditionally always been favoured structure for semantic segmentation issues. Dynamical models and convolutional neural networks (CNN) outperform LSTMs for text categorization tasks, nevertheless. Transformer models are the top performers, but they have the drawback of having a large number of parameters and requiring trained language models. There are just a few pre-trained language models available for various languages. In general, CNNs are not constrained by these issues and have proven to be very effective at text categorization tasks.

When utilizing CNNs for text classification, there are two methods: word-level (word-CNN) and character-level CNNs (char-CNN). Character-level techniques typically perform worse than word-level ones. Word-level techniques, however, need a trained word model. This need has a similar restriction to transformer model limitations. The disadvantage of word-level CNNs is that the input text must be well before in order to

CONTACT C. Maniveena ✉ maniveenac.cse@gmail.com Department of Computer Science and Engineering, Noorul Islam Centre for Higher Education, Thuckalay, Kumaracoil Kanyakumari Dist, 629180 TN

remove regular expressions, stemming words, remove punctuation, and handle and in phrases. Each of these steps raises the risk of input text contamination or accidental deletion of essential distinguishing information. Another potential problem is the lack of a word-model that has been trained for a particular language. For CharCNNs, from before the words or transfer learning are not required.

Moreover, any sources of errors brought on by incorrect pre-processing are reduced because char-CNNs do not entail pre-processing of the input text data. The disadvantage of char-CNNs is that they are typically less accurate than word-level CNNs. According to study, increasing depth to a char-CNN does not result in radically better performance, as seen in the picture categorization domain. A network design called DenseNet has demonstrated strong performance in the picture domain and lately displayed encouraging results in the text classification area. The outcomes, though, have yet to show revelation advancements. There has been little to no additional study on DenseNets used for text classification, and it is unclear other hyper parameters or DenseNet designs can further enhance classification performance. Deep network architecture design, training, and testing are not simple tasks. A network topology has a learning bias, and the effectiveness of the network is based on the calibre of the selected model parameters. Trial and error or the practitioners' experience is used to choose these features. A method of simulated annealing called evolutionary deep learning (EDL) tries to automatically find and implement network designs for a specific goal.

Goal:
The objective of this work is to autonomously generate efficient character-level DenseNet structures for text classification problems using GP and the back propagation algorithm. These goals are accomplished by completing the following goals:

(1) Create a suitable implied for char-DenseNet topologies so that it may be applied to GP-based algorithms.
(2) Examine the suggested application's generalizability across various text classification tasks using two well-known text classification datasets, one of which is small in size and the other of which is huge in size.
(3) Make a comparison of the best-evolved classifiers against the most advanced char-CNN and char-DenseNet models currently available.

2. Related works

Vivek Balachandran et al. (2013) have suggested a strong, covert, and affordable algorithm to disguise software applications. The user-released software code is susceptible to assaults via source code. It is possible to use software obfuscation techniques to make it more difficult to reverse engineer software programs [1]. The algorithm's main goal is to conceal control flow information in the data area by removing it from the code area. The memory-access validation technique by Dongkyun Ahn et al. (2015) contains data about spurious data at the level of the cache line size. In order to enable blocking of erroneous data prior to control flow diversion, a validation unit based on the proposed scheme responds to inquiries from other processor components [2].

Rajesh Sharma et al. (2015) have indicated that the main objective of this study is to discuss cutting-edge research findings and strategies offered for CRN security to safeguard both licensed primary users and illegal secondary users. This paper also discusses the most recent developments in security threats, attacks, and countermeasures in CRNs, with an importance placed on the physical layer. These developments include categorizing security threats and attacks according to their types, existence in the CR cycle, network protocol layers exploited during their activities, defense tactics, and system dynamics methods [3]. By utilizing formal threat models that are represented as Predicate/Transition nets, Dianxiang Xu et al. (2012) have proposed a method for the automated development of security tests. A threats model is used to produce all attack pathways, or security tests, which are then translated into executable test code in accordance with the provided Model-Implementation Mapping standard. A growing number of examples of privacy leakage in Cyber Physical Systems have been explored by Heng Zhang et al. (2016). Our society is quite concerned about the catastrophic implications that result from this. Most privacy-preserving techniques that are offered to guard sensitive personal data also harm system performance. This study analyzes the compromises among system performance and personal privacy in CPS[4].

An analytical method based on stochastic Petri nets has been put out by Robert Mitchell et al. (2016) to explain the interplay between defence and adversary behaviour for cyber physical systems. This essay focuses on several failures that can affect a cyber-physical system, such as attrition failure, pervasion failure, and intelligence gathering failure. The research also suggests an intelligent collaborative security paradigm to reduce security risk, according to Riazul Islam et al. (2015) [5]. In a wormhole attack on a networked control system, as described by Phillip Lee et al. (2014), an adversary creates a connection between two separate geographical regions of the channel using either high-gain transmitters, as in the case of the out-of-band time portal, or cooperating network nodes, as in the case of the in-band time portal [6].

Song Han et al. (2014) have recommended In order to properly apply the intrusion detection technique to CPS, it will be attempted to discuss this in this work. Finally, a few important research issues are noted in order to shed light on the forthcoming studies. In one study, Taheri, Shayan et al. (2018) used the CTU-13 Dataset for evaluation and the Python programming language to create the system. Our model findings show that applying transfer learning can similar comparative from 33.41% to 99.98% [7,8]. Support Vector Machine (SVM) and logistic regression, two more classifiers, have also been applied. They were 83.15% and 78.56% accurate, respectively. In these studies, the algorithm also did a great job classifying the data. Then, we define the aforementioned "forward problem" and "inverse problem" in the context of incident identification in a MIoT [9,10].

We suggest two optimization-based equations with average queuing delay and average delivery rate needs for quality-of-service (QoS) restrictions. We look into both instances of flawless and inaccurate spectrum sensing. We provide three message-decoding algorithms at the relay nodes and evaluate their performance to further improve the users' QoS requirements. To determine which decoding approach may attain that limitation, we calculate an upper bound on the secondary queue average service rate. Our numerical findings demonstrate the advantages of relaying and its capacity to improve both primary and secondary users' performance. Additionally, the proposed methods' success comes very near to the obtained arbitrary limit [11,12].

The theory behind it is that multi-view methods, which are frequently disregarded in the literature, can provide more information for categorization. Semi-supervised learning can be used to make the most of both labeled data. In the evaluation, we look at how well our suggested approach performs using two datasets and a real network setting. Experimental findings show that using multi-view data may classify emails more accurately than using single-view data, and that our approach is superior to several other similar algorithms already in use [13,14]. It is a categorized list of trust- and friendliness-based social IoT techniques that highlights key restrictions like scalability, adaptation, and appropriate network structures (for example, human-to-human and human-to-object) [15]. Additionally, usual issues like social contacts and communities of interest are thoroughly examined, with a focus on kindness and trust-based attributes like task scheduling, social resemblance, and integrated cloud services [16].

After categorization, OHE performs encryption and decryption on sensitive data. The key is verified during encryption, and the best key is chosen using the Step Size Fire Fly (SFF) optimization method. This method can generate the encrypted key and produce the most significant IoT data that protects privacy. The results demonstrate that the proposed IoT security model performs with high security while achieving maximum key breakage time and minimal computing effort [17,18]. Our experiment's high categorization accuracy results offer a promising means of understanding a significant social issue through the usage of major social media platforms and their application to meeting the informational objectives of diverse community welfare groups.

In order to detect threats in the social internet of things, this project aims to use a novel technique to cyber security called deep learning. The findings of the studies show that our distributed detection accuracy technique is superior to centralized detection approaches based on deep learning. In terms of attack detection, it has also been demonstrated that the deep model works better than its superficial equivalent [19].

In order to detect racist statements or meaning, this article applies the necessary machine learning algorithm to a set of papers that contain racist comments. Finding the degree of similarity between a pair of text messages used as a source and terms that are defined as disruptive or in discriminatory terms is necessary to identify antisocial content. This article's method for identifying antisocial conduct is based on a method for classifying material based on phrase frequency [20]. To accomplish this, ML is being utilized as a powerful tool to detect assaults and spot strange network patterns. The architecture of the Internet of Things covered in this survey article which also discusses the significance of IoT security in terms of several forms of potential attacks. Additionally, promising ML-based IoT security solutions have been given forthcoming difficulties explored [21]. The suggested fire detection system combines cloud computing, UAVs, and wireless sensor technology. The suggested fire detection system additionally incorporates computer vision techniques to more accurately and efficiently identify fire occurrence. There are also developed to increase the detection rate. It has been noted that the suggested method has a greater rate of fire detection increasing from 95 to 98 percent [22,23].

Several IoT vulnerabilities are being introduced as the use of IoT devices grows every second. According to the findings and analyses, the IoT paradigm is experiencing new security issues due to the widespread adoption of new technologies [24]. We suggest a system that recognizes various IoT device communities that contain reliable and social relationships. After that, we train a machine learning model to find the total time required for the required functions to be processed by potential candidates belonging to the same congregation as the requester. This algorithm takes into account multiple computational and non-computational attributes of the user as well as the edge computers [25]. PSO is a population-based evolutionary search method that

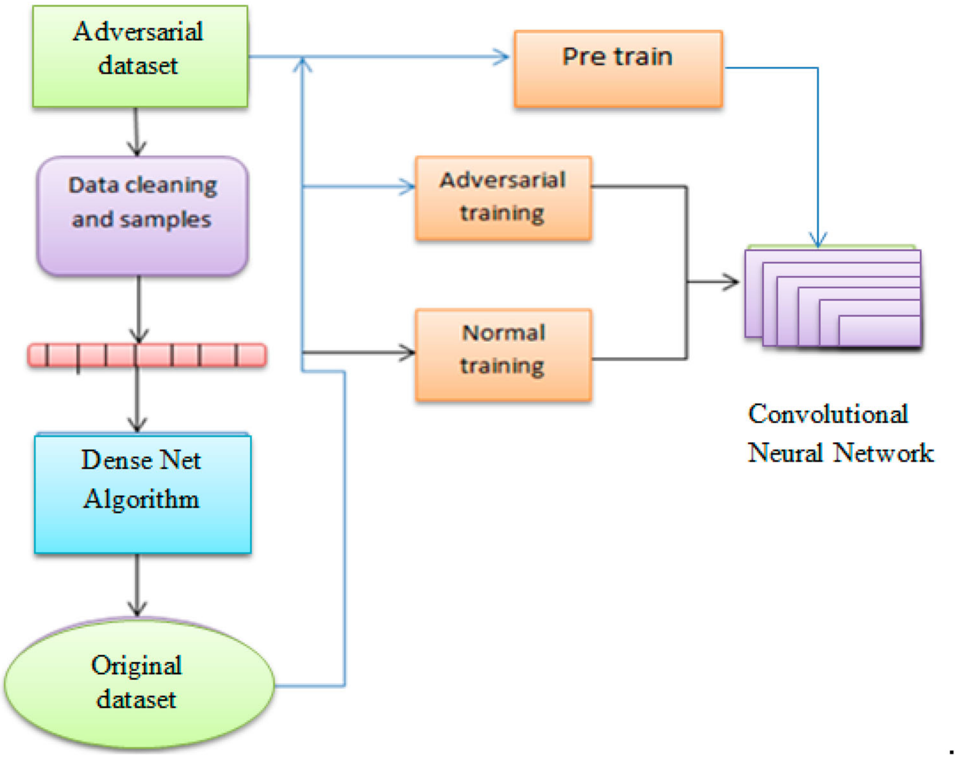

Figure 1. Overview framework of the proposed method.

encodes each layer of CNN architecture according to a decision variable. It is inspired by the social behaviour of animals, such as fish schooling, birds flocking, and mammal herding. Different types of network layers would be correspondingly mapped to distinct subregions inside the decision variable search space. CNN architectures evolve as a result of potential changes to the values of bits related to ID or parameters caused by updates to a decision variable. Thus, as a particle flies to a new place in the search space, it often generates a new CNN architecture [26].

The proposed DMFL_Net model collects data from many hospitals, builds the model with DenseNet-169, and generates reliable predictions from data that is securely stored and shared with only authorized users [27]. When specific consumers share their credentials and claim to have personal experience with risk factors, machine learning algorithms offer a risk reduction strategy that avoids and predicts financial scams, perhaps saving their lives [28].

Objective:
The primary goals of this study are to generate high-quality text from the provided textual captions, to outperform other models in regards to performance and to reduce loss and retraining time for the proposed DenseNet in comparison to earlier findings from other similar efforts.

3. Proposed method

This section introduces the CNN model and provides a detailed description of our suggested security enhancement mechanism. First, the attention mechanism a powerful model of long-term memory in the field is utilized to extract the similarity measurement of the words. We employ the DenseNet technique to generate the countermeasure samples. Instead of just using the original samples for training, we also use the conflict samples in this process. Additionally to ensure that the system is overly near to the training set and to avoidance excessive operating convert layer. And to increase the resilience of our model, adversarial training is employed. Below more information regarding the construction of the semantic similarity analysis model and the development of adversarial samples will be provided. The proposed model framework is shown in below Figure 1.

3.1. *Adversarial convolution neural network model based on attention mechanism*

(1) Sentence Pair Mutual Information Extraction: relationships, and position relations, always have significant effects on sentence semantics when analyzing sentence pair similarity. However, common sentence pair similarity analysis techniques take into account the information that each phrase pair shares with the other. These semantic vectors alone cannot capture the complete information flow. The model's accuracy will thereafter be severely compromised. Our model is based on the mechanics of attention. Without a doubt the model will be even more accurate if the details can be given greater focus during the neural network's extraction of

> sentence features. Therefore, before the pair is entered into the anti-convolution neural network, the suggested model weighs the mutual information of the sentences. Additionally, Word2vec word vector integration and co-occurrence word position information embedding are the major topics. The word is initially preprocessed before features are extracted. Word cleaning, features extraction, word identification, and breaking are all included in the preprocessing. First off training samples for information extraction frequently include a large amount of noise data, including misspelled words, words in languages other than English, and unhelpful phrases. The fine-grained preference variables that were retrieved from this type of noisy data will be impacted.

Therefore, it is necessary to first denoise the evaluation data. Then splitting is removed to feature extraction process that will need to be later analyzed. It involves gaining roots by the removal of affixes. For instance, the root "cat" is used to recognize the strings "Cats", "cat-like", and "catty;" The words "stem", "stemming", and "stemmed" all derive from the word "stem". Finally, because nouns, verbs, and adjectives that are most likely to represent fine-grained qualities, part of the speech-tagged function is used to retrieve these parts of speech from replies.

$$\mathrm{w_2vembedding} = \sum_{i}^{n}\sum_{j}^{m} cos(\mathrm{w_i},\ \mathrm{w_j}') \tag{1}$$

Where the words in the sentence pair are wi and wj. Equation (6) was used to determine the phrase pairing word embedding matrix. Then, using the computation approach stated in formula 7, the weight matrix between sentence pairs is obtained. Adding the row elements of the matrix weight vector for each conceptual unit in a sentence A relative to sentence B is determined. Construct the weight vector of each conceptual unit in sentence B about sentence A by adding the matrix column elements.

$$\mathrm{w2vmatrix} = \sum_{i}^{n}\sum_{j}^{m} cos(\mathrm{w_i}, \mathrm{w_j}') = \begin{bmatrix} \cos(w_1, w_1), \ldots\ldots, \cos(w_1, w_j) \\ \cos(w_2, w_2), \ldots\ldots, \cos(w_2, w_j) \\ \cos(w_3, w_3), \ldots\ldots, \cos(w_3, w_j) \\ \cos(w_i, w_j), \ldots\ldots, \cos(w_i, w_j \end{bmatrix} \tag{2}$$

In addition to semantic similarity, context, and text structure information that Word2vec embedding takes into account, the number of words in a sentence and their relative positions also have an impact on semantic alterations. Based on the shortest path of the text's words, position embedding creates a position-embedded weight matrix. The co-occurrence phrases in the text must first be generally retrieved before the position integrated weight matrix can be created. Then, a set of words with co-occurrences is generated, where k denotes the number of pre in the sentence, and set comword = wc1, wc2 ... wck, set (A) (B). Second, get the word positions knowledge. $\mathrm{w_k}^{\mathrm{c}}$.

3.2. Dense networks

DenseNets attempt to address the gradient problem rather than the gradient degradation problem. DenseNets transfer the outputs of each layer to all layers rather than having identity mappings from one layer to the next they differ from ResNet. It gives each layer access to the aggregate knowledge of all layers before it. As a result, later layers "reuse" the feature maps. Because of the compound nature of DenseNets and the reuse of feature maps, fewer feature maps are needed as input. These feature reuse and identity mappings are applied to each dense block. There are transition layers between each dense block that are made up of convolution and a pooling layer.

Algorithm 2: Proposed Dense Net

```
$\mathrm{SNR_m} \leftarrow$ max (Input)
$\mathrm{SNR_{req}} \leftarrow$ demodulation floor (current datasets)
 Device margin $\leftarrow$10
$\mathrm{SNR_{margin}} \leftarrow$ ($\mathrm{SNR_m}$ –$\mathrm{SNR_{req}}$ –deviceMargin)
Steps $\leftarrow$floor ($\mathrm{SNR_{margin}}$/s)
while steps $>$ 0 andSF̄ $> SF_{min}$ do.
   $SF \leftarrow SF - 1$
   $steps \leftarrow steps - 1$
while steps $>$ 0 and TP $> TP_{min}$ do
   $TP \leftarrow TP - 3$
   $steps \leftarrow steps - 1$
while steps $<$ 0 and TP $< TP_{max}$ do
   $TP \leftarrow TP + 3$
   $steps \leftarrow steps + 1$
end
```

3.3. Dense blocks (cell)

As shown in Figure 2, a dense block (cell) is made up of several convolutional blocks. A convolution operation, a training algorithm, and a fully connected layer make up each convolution block. The input units of each convolutional block that comes after are linked with the output channels in an input manner. A convolutional block's previous input channels are concatenated together. It has been demonstrated that densely connected blocks improve classification efficiency and reduce the vanishing gradient issue. A transitional layer, consisting of a 1 × 3 convolution and a 1 × 2 local max-pooling layer, follows each dense block.

3.4. Classification

A softmax layer with full connectivity makes up the classification layer. Based on the amount of malware classes present in the dataset, FC's number of neurons is set. Multi-class classification issues are categorized using the softmax algorithm. With regard to all

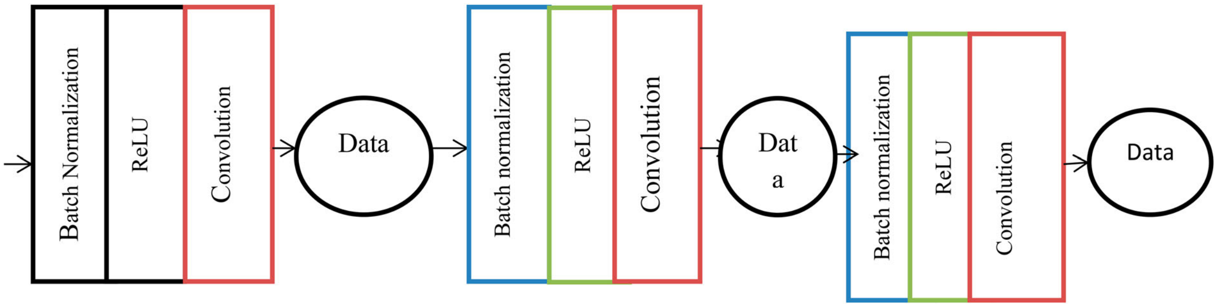

Figure 2. Structure of a dense block's interior (cell).

potential classes, this function determines the posterior distribution for each class. providing the softmax activation function is

$$\mathrm{S}\,(\,y_i) \;=\frac{e}{\sum}yi \tag{3}$$

where y_i is the input value and y_j is all input values of I. The formula calculates the ratio of the exponential of the input element and the sum of the exponential values of all input data.

$$\mathrm{W}_i\acute{\alpha}\ 1/s_{ni} \tag{4}$$

In the case of the class imbalance problem, the distribution of classes in the training dataset is unbalanced. The dominating classes are favoured by models that were built using these various sample sizes. Data augmentation methods like oversampling minority classes or undersampling majority classes are inappropriate for malware detection issues since they do not address the problem of data imbalance. By oversampling, it is impossible to produce images that correlate to realistic malware files. By down sampling, many significant malware variants may be ignored.

where S_{ni} is the effective number of samples for class i. It is given by.

$$S_{ni} = \ (\,1 - \mathrm{B}_i{}^{n})\ /\ (1 - \ \mathrm{B}_i) \tag{5}$$

3.5. Terminal set and decorator set

Dense block parameter terminals (DBT) and Training parameter decorators are described as two sets (TPD). The digits in the DBT set fall into the range of 1–10. The number of convolutional blocks that make up a dense block is determined by this terminal set (cell). On all dense blocks, DBT is utilized (cells). The TPD measures the likelihood that a dense block will be dropped during a batch training phase and is composed of real numbers in the range of 0.0–0.5. In order to avoid the neural network from becoming too dependent on a certain feature discovered from the input data, this number of actions as a dropout component. If a dense block has a dropping value of 0.5, the neural network will skip over it after every previous package of training data. Each symbol in a GP tree can be decorated with a variable from the TPD terminating set to control the dense block dropout behaviour during the training cycle of the neural network. To decide whether or not to design a dense block (cell) with the drop-out feature, a probability value is utilized. This is the only work that, as far as the authors are aware, incorporates a CNN trained hyper component into the genotyping of genetic algorithms.

3.6. Evolutionary operators

To create a reproductive pool, k randomly chosen genotypes are chosen from the population using tournament selection. In this work, one-point crossover is used. At a selected random node in the GP tree of each genotypes, the subtrees of that genotype are traded. This procedure produces two genotypes of offspring. The genetic programming of choice is uniform. By placing a tiny randomly created subtree there, a chosen genotype is modified at a random location in its GP tree structure. The focus of upcoming study will be on innovative mutational and crossover operations.

4. Experimental results

4.1. Datasets

The fourth dataset, the Malicia dataset, was utilized for testing while the MSRP (Microsoft Research Paraphrase Corpus) dataset was used for training. 5100 cleaned ware samples were used in the trials. These samples were gathered from Windows application programs (.exe) and examined through the Virus Total site. The several malware dataset groups that were utilized to evaluate the suggested malware detection approach are listed. Across many datasets, the amount of malware samples varies by type. The dataset comprises 5100 harmful samples that were displayed as greyscale text.

4.2. Evaluation for our comparative analysis, we have used four evaluation metrics

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

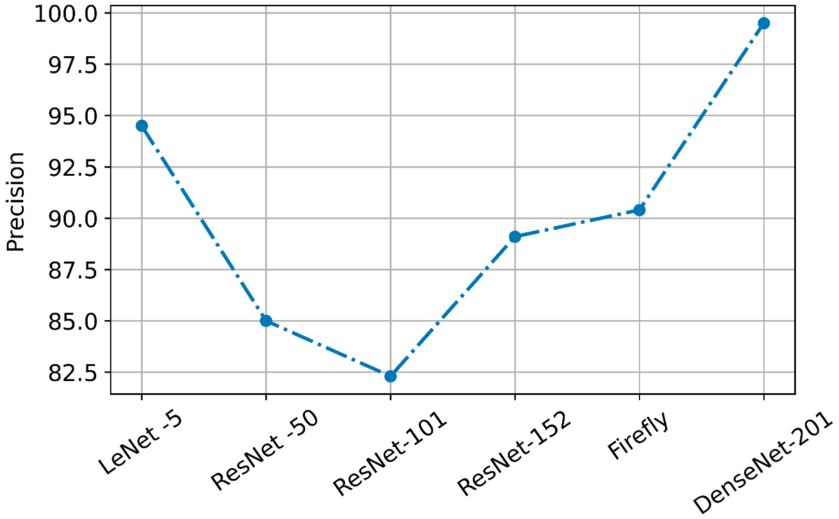

Figure 3. Comparison for different evaluation metrics of precision.

Table 1. Parameters of DenseNet201.

Layer Name	Output Feature Map Size	Kernel Size	Number of Iterations
Image Input	224 × 224 × 3	-	-
Conv 1	112 × 112 × 64	7 × 7 conv	1
Max Pooling	56 × 56 × 64	3 × 3 max pool	1
DenseBlock_1	56 × 56 × 256	1 × 1 conv 3 × 3 conv	6
DenseBlock_2	28 × 28 × 512	1 × 1 conv 3 × 3 conv	12

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

$$\text{F1score} = 2 \times \frac{precision \times Recall}{Precision + Recall} \tag{8}$$

The number of accurately recognized malicious samples is known as True Positives (TP). The number of accurately recognized benign samples is known as True Negatives (TN). The quantity of benign samples labeled as harmful is known as false positives (FP). False Negatives (FN) are samples the model failed to correctly identify. The parameters are shown in below Table 1.

A measurement of accurate classification is accuracy. The ratio of precise positive forecasts to all predictions is known as precision. Precision is crucial because if it's low, the model could incorrectly predict the infection of many healthy samples. If samples from cloud data centres are labeled as malicious, this may reduce the availability of numerous services. Recall measures the proportion of true positives to all actual positives. This statistic is significant because it shows how frequently infected samples slip past the model's detection. When a false negative has a substantial cost, like when discovering malware does, recall is helpful. When there needs to be a balance between Precision and Recall and there is a significant imbalance in the dataset, the F1 score is applied.

Table 2 displays the findings for each CNN model taken into consideration. These numbers were collected from the model that performed the validation data set after each model was tested throughout 100 epochs. This implies that these represent each model's best-case scenario. These models would just be trained until they produce the best results if they used in a cloud environment. The best models should be rather than models that are compared based on random criteria, such as after n epochs, because this point may vary for each model.

Table 2. Results for different evaluation metrics.

Model	Accuracy	Precision	Recall
LeNet −5	88.9	94.5	80.9
ResNet −50	87.4	85.0	88.9
ResNet-101	86.5	82.3	89.6
ResNet-152	92.2	89.1	91.4
Firefly	71.3	90.4	91.2
DenseNet-201	92.8	99.5	91.5

The 5100 data-collecting activities that made up the dataset were divided into three categories: training dataset (80%), validation accuracy (30%), and testing data (30%). The validating and testing datasets were not easily confused, just the training dataset. Dense Nets achieved the maximum accuracy and precision, at about 93% and 100%, respectively. Additionally, with 91.5%, Dense Nets got the highest F1 scores. The best recall score was 89.7% for ResNet-101.

As the harmonic mean of Precision and Recall, F1 accurately depicts the performance of the models. Since the stability of the model cannot be clearly expressed, we decide to make inferences using the particular variable Sec. As a measure of stability, the Sec combines the traits of the two variables F1 and Correct Rate. And denotes the coefficient ranging from 0 to 1. The model's robustness will increase as Sec increases since it will be

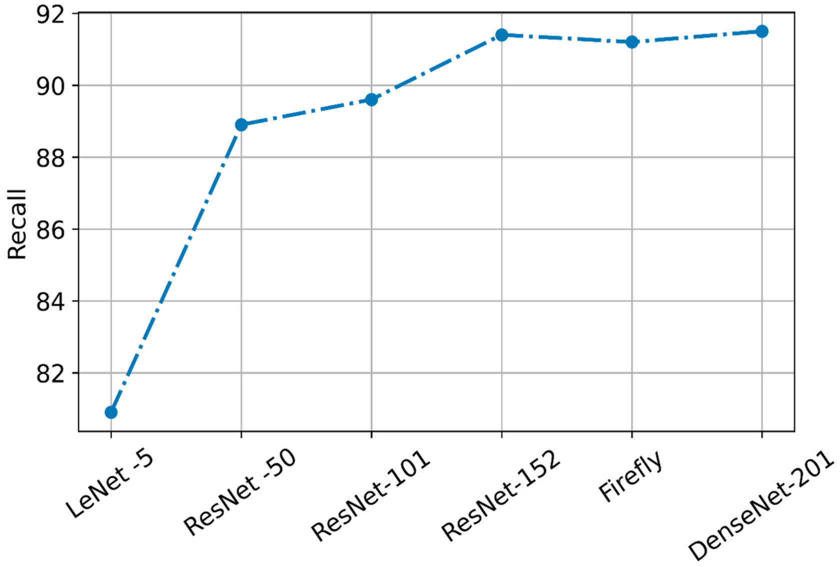

Figure 4. Comparison for different evaluation metrics of recall.

less susceptible to text attacks. The one with a lower Sec value, on the other hand, will be thought to operate worse.

Recall value is a statistical measure of the accuracy of dichotomous models, used to measure the accuracy of unbalanced data. It also takes into account the accuracy rate and recall rate of the classification model. Recall can be regarded as a weighted average of model accuracy and recall rates. Furthermore, we use the indicator Sec to quantify the model's comprehensive performance.

ResNet-5 and ResNet-152 are comparable in effectiveness. Such a minor improvement in accuracy over LeNet-5 may not always be worth it, especially given the somewhat extended training period for ResNet-152. Keep in mind that ResNet can operate better when compared to other parameters and, as a result, might be effective in many circumstances. The least accurate ResNet models are ResNet-50 and ResNet-101. Firefly has a lower score than descent, which is −201. Due to the dense blocks' ability to reuse features, the DenseNets outperformed the other models. Additionally, DenseNet models outperform the competition in terms of feature efficiency.

In this scenario, q = 0.7 is chosen as the likelihood that edge nodes will reject requests from IoT devices, whereas p, 0.42, 0.44, and 0.47 are chosen as the probabilities that IoT devices will submit malicious requests. As seen in Figure 4, there is a declining trend. Notably, it converges to 0 more quickly the lesser the probability of malicious requests from IoT devices, which means the edge nodes are more likely to comply with the requests. Using the instances of $p = 0.42$ and $p = 0.47$, the former falls to 0 in about a half-game whereas the latter requires the third game to do so. When p is positive, it is inferred that the granted request is the transformation edge node approach $< \frac{\beta\gamma+\varepsilon}{2\pi\tau\epsilon-a\delta\epsilon+\beta\gamma+\epsilon p}$

Table 3. Payoff matrix.

IOT devices	Edge notes Detect & Grant (DG)	Detect & Deny (DD)
Request Maliciously (RM)	$\mathbf{(1\text{-}\alpha)\delta\xi_D + \varepsilon\xi_A}$ -∝ $\boldsymbol{\xi_S}$ –ÇD, ξ_S-(1-α) $\delta\xi_D - \varepsilon\xi_A$ –ÇS	$\delta\xi_D$ -∝ξ_S-Ç$_D$ $\delta\xi_S + \epsilon\xi_A - \delta\xi_D$ –Ç$_S$
Request Normally (RN)	ξC + ϱ –ÇC, ξ_p –ÇS	ξ $_C$ + ϱ –ÇC -βy-ÇS

The anticipated profitability of IoT devices sending fraudulent queries is shown in Table 3 as follows:

$$\begin{aligned} E(RM) &= q((1-\alpha)\delta\xi_D + \varepsilon\xi_A - \alpha\xi_S - \zeta D) \\ &\quad + (1-q)(\delta\xi_D - \alpha\xi_S - \zeta D) \end{aligned} \tag{9}$$

and also the following is the predicted income from IoT devices completing typical requests:

$$\begin{aligned} E(RN) &= (\xi_C + \varrho - \zeta c) + (1-q)(\xi_C + \varrho - \zeta c) \\ &= \xi_C + \varrho - \zeta c \end{aligned} \tag{10}$$

5. Limitations and challenge

Though our findings show which CNN model performs better in certain scenarios, there are several restrictions we like to draw attention to base on our own experience. CNN's inability to recognize an effect is particularly in the data set is its most significant drawback when applied to the kind of data we used. It is crucial for a model to have some understanding of the current sampling and the behaviour of the computer over time when detecting malware in a virtual machine that is already up and running. One such instance is when a machine experiences an increase in traffic, and because of a scaling limitation, the samples produced by that machine start to resemble some harmful samples. In this situation, the false positive rate can rise if the model

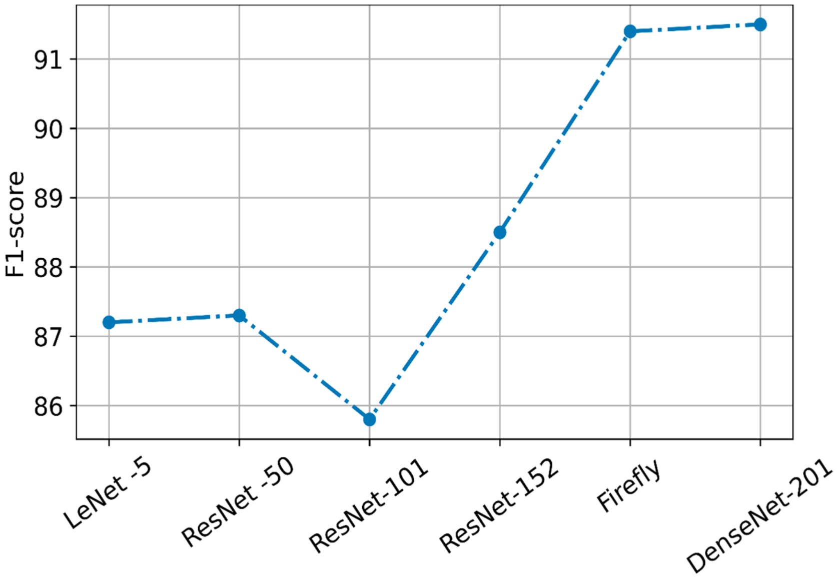

Figure 5. Metrics comparison for used CNN models of F1-score.

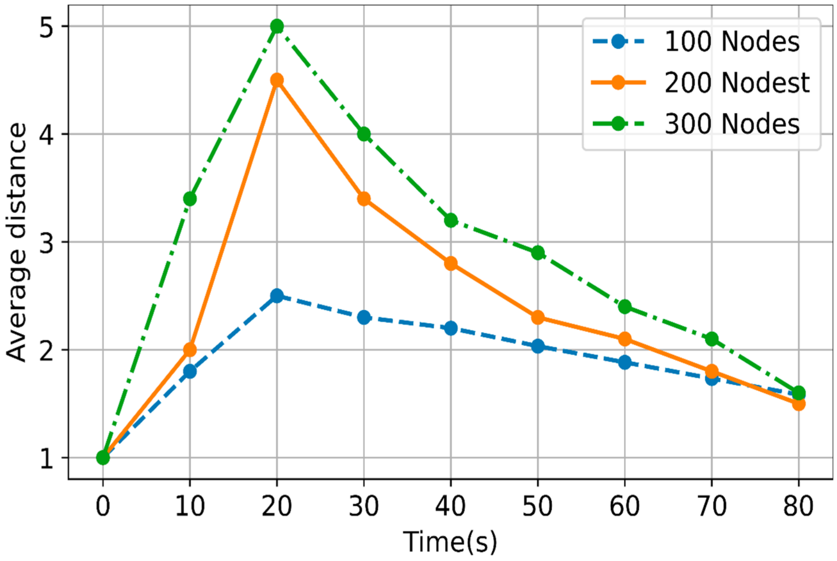

Figure 6. Edge node propagation shapes: choosing an approach when $p < \frac{\beta\gamma+\varepsilon}{2\pi\tau\epsilon-a\delta\epsilon+\beta\gamma+\epsilon p}$.

does not learn that process measurements can be modified in accordance with core purpose on the machine. Another possibility is that the malware identifies an infected sample and instantly goes latent to conceal itself. The likelihood of false negatives, where the model misses a malware infection even while it is disguised, may rise if the model does not account for the preceding sample where the malware was discovered. As a result, it may happen that samples are mistakenly classified as dangerous or benign. This issue was solved in however it is unlikely that all samples would be properly labeled without developing customized malware that will signal when dangerous behaviour begins and finishes.

6. Conclusion

In order to decrease loss and training time and produce text with more distinctive characteristics, this research suggests using the Dense Net architecture. In comparison to the current CNN design, our method was able to cut loss by 1.62% and obtain results 768 s faster per cycle. This work proposed an evolution deep learning approach to automatically produce feature DenseNet frameworks for text classification problems (GP-Dense). This goal was successfully accomplished through the development of GP-based genetic algorithms using an indirect decoding to characterize probable neural network configurations. Using only 25% of the available datasets, GP-Dense was able to successfully manufacturing method network architectures. On the smaller dataset, AG News, GP-Dense fared better than it did on the larger sample, Reviews Full. This work demonstrates that EDL has the capacity to perform well on growing character-level DenseNet topologies for text categorization problems. It has shown, however, that additional study is necessary to ensure that performance architectures can be improved for larger datasets, including tweaking constants and using a larger portion of a training dataset.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

[1] Balachandran V, Emmanuel S. Potent and stealthy control flow obfuscation by stack based self-modifying code. IEEE Trans Inf Forensics Secur. 2013;8(4):669–681.

[2] Ahn D, Lee G. A memory-access validation scheme against payload injection attacks. IEEE Trans Dependable Secure Comput. 2014;12(4):387–399.

[3] Rajesh K, Sharma, Danda B. Advances on security threats and countermeasures for cognitive radio Networks. A survey IEEE communications surveys & Tutorials, ISSN: 1553-877X. 2014.

[4] Xu D, Tu M, Sanford M., et al. Automated security test generation with formal threat models. IEEE Trans Dependable Secure Comput. 2012;9(4):526–540.

[5] Zhong H, Wen J, Zhang S. Efficient conditional privacy-preserving and authentication scheme for secure service provision in VANET' tsinghua science and technology. 2016.

[6] Ramesh MRR, Reddy CS. A survey on security requirement elicitation methods: classification, merits and demerits. Int J App Eng Res. 2016;11(1):64–70.

[7] Jensen R, Shen Q. Fuzzy-rough sets assisted attributes selection. IEEE Transactions on Fuzzy Systems, ISSN: 1941-0034. 2007

[8] Lee P, Kezunovic M. 2016, Fuzzy logic approach to predictive risk analysis in distribution outage management. IEEE Transactions on Smart Grid. 2014.

[9] Taheri S, Salem M, Yuan J-S. Leveraging image representation of network traffic data and transfer learning in botnet detection. Big Data Cogn Comput. 2018;2(4):37. doi:10.3390/bdcc2040037

[10] Cauteruccio F, Cinelli L, Corradini E, et al. A framework for anomaly detection and classification in multiple IoT scenarios. Future Gener Comput Syst. 2021;114:322–335. doi:10.1016/j.future.2020.08.010

[11] Hussain F, Hussain R, Hassan SA, et al. Machine learning in IoT security: current solutions and future challenges. IEEE Commun Surv Tutorials. 2020;22(3):1686–1721. doi:10.1109/COMST.2020.2986444

[12] Li W, Meng W, Tan Z, et al. Design of multi-view based email classification for IoT systems via semi-supervised learning. J Netw Comput Appl. 2019;128:56–63. doi:10.1016/j.jnca.2018.12.002

[13] Amin F, Ahmad AS, Choi G. Towards trust and friendliness approaches in the social internet of things. Applied Sciences. 2019;9(1):166. doi:10.3390/app9010166

[14] Kalyani G, Chaudhari S. An efficient approach for enhancing security in Internet of Things using the optimum authentication key. Int J Comput Appl. 2020;42(3):306–314. doi:10.1080/1206212X.2019.1619277

[15] Subramani S, Vu HQ, Wang H. [IEEE 2017 4th Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE) - Nadi (2017.12.11-2017.12.13)] 2017 4th Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE) - Intent Classification Using Feature Sets for Domestic 129–136. 2017.

[16] Diro AA, Chilamkurti N. Distributed attack detection scheme using deep learning approach for Internet of Things. Future Gener Comput Syst. 2018;82:761–768. doi:10.1016/j.future.2017.08.043

[17] Chandra N, Khatri SK, Som S. Anti social comment classification based on kNN algorithm. In 2017 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), IEEE; 2017. p. 348–354. doi:10.1109/ICRITO.2017.8342450

[18] Tahsien SM, Karimipour H, Spachos P. Machine learning based solutions for security of Internet of Things (IoT): a survey. J Netw Comput Appl. 2020;161:102630. doi:10.1016/j.jnca.2020.102630

[19] Sharma A, Singh PK, Kumar Y. An integrated fire detection system using IoT and image processing technique for smart cities. Sustain Cities Soc. 2020;61:102332. doi:10.1016/j.scs.2020.102332

[20] Sadique KM, Rahmani R, Johannesson P. Towards security on internet of things: applications and challenges in technology. Procedia Comput Sci. 2018;141:199–206. doi:10.1016/j.procs.2018.10.168

[21] Khanfor A, Hamadi R, Ghazzai H, et al. [Ieee 2020 IEEE 63rd international midwest symposium on circuits and systems (MWSCAS) - Springfield, MA, USA (2020.8.9-2020.8.12)]. 2020.

[22] Slabicki M, Premsankar GD, Francesco M. [IEEE NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium - Taipei, Taiwan (2018.4.23-2018.4.27)] NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium - Adaptive configuration of lora networks for dense IoT deployments., (), 1–9. 2018. doi:10.1109/NOMS.2018.8406255

[23] Liu Q, Xiang X, Qin J, et al. Coverless image steganography based on DenseNet feature mapping. EURASIP J Image Video Process. 2020;2020(1):39. doi:10.1186/s13640-020-00521-7

[24] Kim D-Y, Kim S, Hassan H, et al. Adaptive data rate control in low power wide area networks for long range IoT services. J Comput Sci. 2017;22:171–178.

[25] Augustin A, Yi J, Clausen T, et al. A study of LoRa: long range & low power networks for the internet of things. Sensors. 2016;16(9):1466. doi:10.3390/s16091466

[26] Li Y, Xiao J, Chen Y, et al. Evolving deep convolutional neural networks by quantum behaved particle swarm optimization with binary encoding for image classification. Neurocomputing. 2019;362:156–165. doi:10.1016/j.neucom.2019.07.026

[27] Malik H, Naeem A, Naqvi RA, et al. Dmfl_Net: a federated learning-based framework for the classification of COVID-19 from multiple chest diseases using X-rays. Sensors. 2023;23(2):743. doi:10.3390/s23020743

[28] Deepa N, Udayakumar N, Devi T. "A novel two-way cross-tab machine learning approach for predicting life insurance using bivariate exploratory analysis." In 2023 International Conference on Computer Communication and Informatics (ICCCI), pp. 1-5. IEEE, 2023.