SemantJournals

American Journal of Technology Advancement 2997-9382, Vol 03, No. 03, 2026

Article

Dimensionality Reduction for Network Intrusion Classification Based on Dolphin Mating Algorithm

Ahmed Mahmood Khudhur

Artificial Intelligence Department, College of Computer Science and Information Technology, University of Kirkuk, Kirkuk, Iraq

Email: Dr.ahmedm@uokirkuk.edu.iq

Abstract: The rapid rise of the internet and other associated technologies, along with other things, has caused a huge increase in the sharing of sensitive information across multiple networks. Because of this, fraudsters are getting better at their attacks, which shows how important it is to have strong cyber security. Any network security plan needs to have intrusion detection systems (IDSs) as a key feature. IDSs let experts and administrators keep an eye on possible threats by watching network data for signals of suspicious or hazardous activities. Anomaly-based systems employ machine learning (ML) to find suspicious activity in network traffic. This work suggests a feature selection (FS) structure for building IDSs based on anomalies by combining heuristic and ML methods. This proposed study aims to enhance IDS performance regarding attack detection and detection time by employing heuristics to select attributes. In the proposed approach, Dolphin Mating Algorithm (DMA) will be employed to implement the FS mechanism and a customized convolutional neural network (CNN) deep learning model will be implemented for the classification job. The adopted dataset is the NSL-KDD. Out of 41-features that consisting the NSL-KDD dataset, only 9-attributes were selected, in other words, 21.95% of the features will be used to train the CNN model. More than 90% accuracy for the four classes of the attack types was achieved.

Keywords: Dimensionality Reduction, Feature Selection, Dolphin Mating Algorithm, Intrusion Detection System, NSL-KDD dataset, Convolutional Neural Network.

Introduction

Because more people are using the web, important, private, and confidential personal and corporate information is now swiftly spreading throughout the world. Because network threats are always changing and computers are sharing more data than ever, network and information security is more vital than ever. By 2022, global IP traffic is expected to climb by 26% each year, or three times as much. Predictions say that by 2022, the amount of commercial traffic per person per month will have grown from 16 GB in 2017 to 50 GB [1]. It is now evident that this volume is much bigger than expected. The instruments used for gathering data, sharing information, and keeping an eye on

networks make the attack surface bigger. Attackers that want to get into a network's security want to get around data privacy and access by taking advantage of weaknesses. So, cyber security is a field that offers important ways to protect against cyber threats and save money. One of the greatest components of the security construct is the Intrusion Detection System (IDS). In this respect, the Intrusion Detection System can be able to identify different forms of intrusions. The Intrusion Detection System helps network managers keep up with the latest threats by monitoring the network traffic. This helps them identify any unusual or destructive acts or policy violations that may be occurring in the network. This need has led to the development of a reliable IDS that can protect the network against malicious intrusions. The Intrusion Detection Systems have evolved in an effort to make the network traffic secure against hackers and other security threats. Combination of the IDS capabilities that use signature analysis, behavioral analysis and identification of malicious behaviors has assisted the next generation firewalls to counteract complexity of network threat. An IDS will send out alerts in case of detection of malicious or suspicious activities in the network to the concerned parties. To make sure that the system is not accessed and used to carry out destructive activities, intrusion detection systems mainly concentrate on the particular sequences of activities [2].

IDSs are generally categorized as either host-based IDSs or network-based IDSs depending on the deployment situation, place, and sources of data. A host based IDS is able to detect suspicious activity or a breach of security standards by observing the data packets [3]. The downside of this method is that the IDS system will have to be installed in every device and this could create performance issues due to higher processing loads. Conversely, network-based IDS systems intercept, observe, and study network traffic across diverse sources with an aim of identifying malicious activities. IDSs can analyze in two major ways which are the detection based on anomaly and the exploit based detection that is also referred to as signature based detection. The signature-based IDSs identify malicious activities by comparing the information gathered in a network with pre-existing patterns or signatures of attacks. As soon as it rediscovers a pattern, it marks such an action as malicious. The system is suitable in identifying the current attacks, however, it is not capable of identifying new attacks because it can only apply current signatures in its database.

It is also very power-consuming to maintain a signature database and match the database with network data. On the other hand, IDSs that use anomalies to detect intrusion and not the normal operation of the system are referred to as behavior-based IDSs. The ability to identify new threats that are not foreseen is one of the greatest advantages of this approach. This will increase the number of false positive results, which in turn can limit its usefulness in certain cases. Intrusion detection (ID) is one such vital and dynamic discipline which is giving rise to developments that enhance computer and network security as more sophisticated and varied threats to the networks emerge [2]. To develop effective IDS methods, scholars have been conducting various experiments. Besides the fact that most contemporary IDS are signature-based, they are also based on anomaly-based methods supported by AI, namely, machine learning (ML) and deep learning (DL) algorithms, to identify attacks.

In the context of cybersecurity, ML approaches are increasingly being significant to systems such as IDSs to enhance detection of threats, reduce false positives and adjust to new forms of attacks. Although numerous studies on the notion of enhancing the accuracy of ML models exist, little has been done to address the question of the influence of the redundant and uncorrelated nature of large datasets on the performance of the systems. Due to this fact, the computational load is greater and the generalization and overlearning abilities of the models are restricted. Out of the numerous tools that can be employed, feature selection (FS) algorithms can be viewed as one of the most efficient solutions to the said problem. Some progress has been made in the literature about applying meta-heuristic algorithms to FS and IDSs over massive data, however, generally, there has been no research within this area [4], [5]. The integration of IDS into the FS process would ensure that this performance is optimized which has been illustrated in this study through applying a meta-based heuristic method termed as Dolphin Mating Algorithm (DMA). It seeks to fill the gap in this area.

The aim of this project is to classify assaults in datasets with the adoption of ML algorithms. This objective was carried out using the NSL-KDD data that have attracted many research publications. In the case of FS, meta-heuristic algorithms such as DMA are being used. It is followed by a Convolutional Neural Network (CNN), which found in many applications such as those in [6][7][8][9][10], model to achieve classification once the datasets have been processed with the aid of FS based on the ML approaches. The rest of the paper provides in-depth literature research, the description of what IDSs are, the description of the methods and the materials used, as well as a summary of the experimental studies along with their findings. In order to prepare the environment to the current study, the literature review constitutes the brief description of the previous work in the field. In order to gain more insight into the way the experiments were conducted, datasets, algorithms, and tools that had to be utilized are described in the materials and methods section. Finally, there is the experimental investigations and results section that offers the analysis and discussion of the experimental results that can be used to improve the IDS technology and its use.

Related Works

In order to optimize FS and enhance the effectiveness of IDSs, Salek et al. performed a study using the differential evolution (DE) algorithm in conjunction with a neural network (NN) classifier. Their method achieved an impressive 96% accuracy rate when tested on the KDDCUP99 dataset [11]. To enhance IDSs, Popoola and Adewumi [12] used the DE algorithm in conjunction with a Decision Tree (DT) classifier to fine-tune FS. They achieved an accuracy rate of 89% while testing their approach on the NSL-KDD dataset.

Hajisalem and Babaie developed a hybrid classification system that used Artificial Bee Colony (ABC) and Artificial Fish Swarm (AFS) algorithms [13]. This approach was evaluated using the UNSW-NB15 and NSL-KDD datasets. The Fuzzy C-Mean Clustering (FCM) and Correlation-Based FS (CFS) methods were used to remove redundant and superfluous features respectively to the training dataset, thereby enhancing the efficiency of the training dataset. In order to train the model and to build the if-then rules to discriminate the normal and abnormal records, Classification and Regression Tree (CART) technique was employed, which is based on the attributes that were selected. The evaluation results displayed an impressive accuracy of 99.9%, false alarm of 0.01 percent and detection rate of 99.99 percent. Moreover, the proposed method was also found to be as computationally inexpensive, complex and runtime as existing methods.

Mazini et al. proposed a hybrid method [14] that is a combination of the ABC algorithm and AdaBoost. In the case of FS, the ABC algorithm was applied, and in classification, they resorted to the AdaBoost. The method achieved 98.9 percent accuracy on NSL-KDD and ISCXIDS2012. Further studies based on such data sets were applied in order to compare the proposed approach with others. Adhi Tama et al. proposed one of the approaches of FS that involved ensembles of two-level classifiers [15]. To use training datasets that are smaller in size, procedures such as genetic algorithm (GA), particle swarm optimization (PSO) and the ant colony algorithm (ACA) were employed. The REPTree method was used to find attributes that maximize the performance with regard to classification. Then, an ensemble of classifiers based on bagging and rotation forest was used. The proposed method was tested using the NSL-KDD data (85.8% accuracy rate) and the UNSW-NB15 data (91.3% accuracy rate). According to the study, the use of advanced methods with fewer features enhanced the classification.

Meftah et al. proposed a two-stage anomaly-based IDS using the UNSW-NB15 dataset [16]. First, random forests and recursive feature elimination were used to select the most significant features. The second stage used logistic regression, SVMs, and gradient boosting machines for binary classification to detect malicious traffic. The overall accuracy was improved to 86.04% by combining the SVM classifier's 82.1% accuracy with the hybrid two-stage classification approach.

The problems related to high attribute count such as feature redundancy and curse of dimensionality, which is an imbalance in data space, were discussed in [17]. They proposed an improved feature engineering model based on a novel dual-strategy technique for restoring

balanced symmetry and achieving more comprehensive feature representation. A symmetrical FS method is first introduced by merging an enhanced DMA with the Maximum Relevance-Minimum Redundancy measure. This method not only selects an ideal subset of highly relevant features but also finds a subset of attributes that are both complimentary and redundant. Second, to mine useful information from these duplicate attributes, a feature reconstruction approach based on ensemble learning is proposed. This method creates data symmetry with the properties of interest by aggregating previously uncorrelated pieces of data into one synthetic attribute. Finally, the improved feature engineering approach builds a high-performance feature space by symmetrically merging the synthetic attribute with the best attribute subgroup. The model's excellent performance has been rigorously validated on 9-typical UCI regression datasets. It obtains an average goodness-of-fit of 0.9263 compared to other methods and exceeds them by a wide margin; results from the Wilcoxon signed-rank test indicate that this increase is statistically significant. Further validation comes from ablation studies as well as detailed assessments regarding parameter sensitivity, resilience, convergence, and latency. This augmented feature engineering approach has been applied to real-world product demand prediction assignment and proved its practical relevance in complicated settings though at the expense of high computational complexity.

Contactless fingerprint (CLFP) identification is also another application of DMA, a rapidly emerging technology that is potentially more hygienic and comfortable to users compared to traditional touch-based systems without jeopardizing security [18]. Thus, the paper [18] provides a CLFP identification system that uses the DMA, a nature-inspired approach that is appropriate when handling complex optimization problems. Automatic image features are determined in the Histogram of Oriented Gradients (HOG) method and FS process is optimized by DMA. In other words, the use of FS purpose by DMA was also present in this scenario. They used DMA and SVM together in order to come up with a hybrid (DMA-SVM) which takes advantage of the global search through DMA and the reliable classification of the SVM in order to make it more accurate in prediction. The other hybrid model that combines Neutrosophic C-Means (NCM) with DMA-SVM, and third with Fuzzy C-Means (FCM) with DMA-SVM are also proposed. Experimental verification of 504 CLFP pictures of Hong Kong Polytechnic University dataset has shown a significant trend of achievement: DMA (91.0%), DMA-SVM (94.07%), FCM-DMA-SVM (96.03%), and NCM-DMA-SVM (98.00%). The NCM-DMA-SVM approach also defeats the rival due to its neutrosophic logic in dealing effectively with ambiguity as well as its accuracy and computational efficiency. According to a comparative study with the current bio-inspired systems, their methodology is more accurate with lower processing requirements. Integration of bio-inspired optimization and traditional classifier with advanced clustering in identification of biometrics has shown positive outcomes as observed here.

DAMA is also used in the echolocation system. Majority of the technology that man has invented may be dated back to things that are found in nature. Dolphins are considered second smartest animals after humans, by the scientific community. Echolocation, a sort of biological sonar, is used by dolphins and other animals to locate their direction and find prey in various environments. Writers in [19], they are postulating a novel optic procedure that follows the examples of dolphins capacities. Although it is possible to find various meta-heuristic optimization methods, scholars have been striving to streamline the process of parameter tweaking as it eats up so much time on the part of the user. According to studies, one can improve results by being aware of the laws that govern meta-heuristic algorithms. Dolphin echolocation has only a few configuration options, which it uses to perform better than most of the existing optimization methods. The new approach can generate quality results at the low computing cost level.

Based on the previous studies, DMA shows significant performance in terms of dimensionality reduction/feature selection process. However, the author did not find in the literature a combination of DMA with a classifier that is based on DNN such as convolutional neural network models. Hence, the presented work will integrate the FS-algorithm that is based on DMA method with a customized DNN built around CNN model. As stated in the abovementioned literature review, NSL-KDD

dataset was best fit for this study. The work is not to identify an attack, but it classify the attacks themselves inside an IDS.

Methodology

The methodology used to conduct this study aims to build a highly accurate anomaly-based intrusion detection system by combining the feature selection methods and a deep learning based approach. In the beginning, we make use of NSL-KDD dataset as the standard dataset which consist of 41 features that describe each type of network traffic and different attack categories such as DoS, Probe, U2R and R2L. The dataset is then pre-processed in order to make it ready for analysis, then divides the dataset into training and test datasets. In order to solving the problems of high dimensionality and redundant features, we apply a meta-heuristic optimization method, namely the Dolphin Mating Algorithm (DMA) for feature selection. By mimicking the social and hunting behaviors of dolphins, the DMA explores the search space to determine the most informative subset of features and simultaneously keeps the feature set very small, whilst maintaining the discriminative power of the initial feature set. The fitness function balances because it preserves classification accuracy while reducing the features ensuring optimal selection. Once the subset of features are chosen it is used to create reduced dimensional input and perform classification task CNN model one customized for the dataset. The CNN architecture is composed of a combination of multiple Convolutional / Batch Normalization / Pooling / Dropout / and Fully Connected layers to learn the complex pattern from the network traffic dataset. Hyperparameters such as learning rate, batch size and number of epochs are optimized and the model is trained and validated using a part of training data. Lastly, the trained model is tested on the test dataset with performance metrics like accuracy, precision, recall, and F1-score to measure the effectiveness of the model in detecting and classifying the network intrusions.

Dataset Description and Analysis

Regarding measuring the efficacy of the ML models, the NSL-KDD is the gold standard in the sphere of cybersecurity and IDSs. This data is a critical resource to scholars and data miners in the area of IDS as it corrects the limitations and biases of the initial KDD Cup 1999 dataset. Owing to the lack of publicly-released datasets of network-based IDSs, it can be held that this new KDD dataset could prove a valuable reference to scholars who compare the different IDSs, although this updated dataset is no longer than an ideal representation of real networks [20]. Moreover, the training and testing sets of NSL-KDD contains relatively large amounts of records. Due to this advantage, the trials on the entire collection are cost-effective as compared to randomly selected subset. That is why you can expect uniform and similar results of evaluation by different studies. This dataset consists of four subdata sets namely: KDDTest +, KDDTest-21, KDDTrain + and KDDTrain +20Percent. Yet, KDDTest-21 and KDDTrain+20Percent are in fact, subsamples of KDDTrain+ and KDDTest+ respectively. In this case, terms train and test will refer to KDDTrain+ and KDDTest+, respectively. A subgroup of the test set named as KDDTest-21 excludes the hardest traffic records (the score of the test is 21), and a subgroup of the train set named as KDDTrain+20Percent includes traffic records that comprise 20 percent of the entire train set. With that said, the traffic records observed in KDDDTest-21 and KDDTrain + 20Percent are not recorded records that are present in either of the data sets, they are already present in the test and train, respectively.

The four types of attacks in the dataset are Denial of Service (DoS), Probe, User to Root (U2R) and Remote to Local (R2L) assaults. Here you may see a brief summary of every attack. The goal of a DoS attack is to terminate all traffic in and out of the attacked system. The IDS will shut down to protect itself once it is flooded with an abnormally large amount of traffic. This prevents normal traffic accessing a network. This can occur e.g. when a well-known online store receives more orders online on the day of one of their big sales and the network collapses as a consequence, preventing all the purchases by the honest buyers. This attack is the most prevalent among the data set. A probe or surveillance attack seeks to have access to data on a network. The goal here is to take advantage of the situation and steal sensitive information, including the names and addresses of customers, or

their financial information. A user-to-root (U2R) attack is also initiated by a normal user account and tries to gain root privileges so that it can attack the system or network. To gain root privileges or access, the attacker attempts to exploit security holes in the system. R2L is a type of attack aimed at remote computers in a bid to achieve local access. A non-user tries to crack into the network or the system without being physically in the same location.

Statistically, NSL-KDD dataset can be described as: the train file contains 125,973 recodes, of them there are 67,343 recodes as normal samples, 45,927 records as DOS attack samples, 11,656 records as Probe-attacks, 995-R2L attack records, and 52-U2R attacks. The test set file involves: 22,544 total records, of them, there are 9710 normal records, 7458-records as DOS-attacks, 2422 records as Probe attacks, 2887 attack records of type R2L, and 67-U2R attack records. Thus, the features can be divided into four-groups, basic-features (from 1-to-9: duration, protocol type, service, and flag), content-features (from 10-to-22: that includes packet data which can be used in R2L and/or U2R attacks), temporal-features, i.e., features that depends on time (from 23-to-31: such as the count-feature and the srv count-feature), and the features that depends on the host itself (from 32-to-41: such as dst host count and dst host srv count features). Nevertheless, employing all of these features will overwhelm the IDS system. Moreover, some of these features will not provide useful information to the intruder identification system or the attack classification system. Hence, some features should be ignored/eliminated through the FS-algorithm, which will be DMA in this paper, as will be seen later.

Proposed Feature Reduction Approach

DMA is a novel approach that draws inspiration from the cooperative communication and hunting strategies used by dolphins [21]. Due to its effectiveness in resolving complicated difficulties across domains, DMA—a method that draws inspiration from nature—has garnered considerable interest. In particular, the algorithm takes advantage of dolphins' complex communication skills and cooperative hunting strategies to benefit from their innate intellect, speed, and social tendencies. A position update mechanism, an equation for speed changes, and the use of echolocation to determine the optimal options are the major aspects of the algorithm [18]. Because of these factors, DMA is a strong and flexible method of solving many different optimization problems, since it may settle to global optimums while preventing local pitfalls. Therefore, to begin, one have to find the dolphin population throughout the search field at random, much like real dolphins do when they spread out to catch prey. Dolphins revise their speeds during the pursuit phase [18],

$$F_{i,j}^{a+1}(p+1) = \gamma \cdot F_{i,j}^{a}(p) + \sigma_1 \cdot \tau_1 \cdot \left(\left(\mu_{best,i,j}^{a}\right) - z_{i,j}^{a}(p)\right) + \sigma_2 \cdot \tau_2 \cdot \left(\Omega_{best,i,j}^{a}\right) - z_{i,j}^{a}(p) \quad (1)$$

where the position ($z_{i,j}^{a}(p)$) will be updated as,

$$z_{i,j}^{a+1}(p+1) = z_{i,j}^{a}(p) + F_{i,j}^{a+1}(p+1) \quad (2)$$

In the last two expressions, the velocity is expressed by $F_{i,j}^{a}$, which is the ath dolphin among its dimension (j) during the i^{th}-iteration, $\mu_{best,i,j}^{a}$ denotes the best position of the dolphin, while $\Omega_{best,i,j}^{a}$ is the best position globally among the entire swarm. Moreover, there are two-acceleration factors, σ_1 and σ_2, and there is an inertia weight coefficient that is denoted by γ. Last but not least, there are two uniformly distributed in the range zero to one random numbers, $\tau_1$and τ_2. As part of their beating phase, dolphins try to close the gap before diving on schools of sardines. After a long period of pursuing the sardine school, these dolphins either swim up to the swarm or swim to a place where they may attack from a good angle.

But we assume that dolphins assume a spherical position in swimming only when they are attacking sardine schools or feeding. When a group of dolphins patrols the same region at different times p(p<M) then some of them will finally find themselves in a position where they can attack or feed on the other group. Accordingly, positions will be updated [18],

$$z_{i,j}^{a+1}(p+1) = \tau_{r,k} \tag{3}$$

and

$$z_{i,j}^{a+1}(p+1) = \tau_{i,k} \tag{4}$$

in which, $\tau_{r,k}$ is the uniformly distributed random number in the range [0, 1] just like $\tau_{i,k}$, where k is the index of a random dolphin in the population, where p is the immediate/current-time step and M stands for the maximum iteration number that should be reached. At last, during the process of changing swimming styles, dolphins enter a dynamic swimming style that lets them swim in a variety of configurations. Currently, we determine the value of (B) by [18]:

$$B_j = \frac{1}{2 \cdot z_{i,j}^{d}(p)}, \quad d = 1,2,\cdots \emptyset \tag{5}$$

In which, the total number of dolphins is represented by $\emptyset$. Assuming that the rank of the B_j values, after ascending sorted, is $R_d(p)$, then the effectiveness of the switching can be determined as [18],

$$E_{i,j}^{d}(p) = \frac{R_k(p) - 1}{\emptyset} \tag{6}$$

Consequently, the selected features that will be adopted by the above algorithm will be utilized in the classifier of the attacks, as will be described in the next section.

Suggested Convolutional Neural Network Classifier

The convolutional neural network will be adopted in this work to base the suggested classifier on. That is, the deep neural network of this work will be consisting of different layers of CNNs. That is, the proposed DNN model has three CNN layers, five batch normalization layers, two max-pooling layers, five drop out layers, a global average pooling layer, and three fully connected or dense layers, where the last dense layer will be the classification outputting layer. The structure can be configured in Figure 1. There are 128, 256, and 512 filters in the first, second, and third convolutional layers, respectively, where the filter size was fixed to 3 in all of the layers. Rectified Linear Unit (ReLU) layer was the activation function for these layers. The pool size of the Max Pooling layers was set to 2. The first and second dropout layers have dropping out probability of 20%, while the third and fourth dropout probabilities are 30%, each, and the last dropout probability is set to 20%. The first and second dense/fully connected layers have 256-units and 128-units, respectively, while the last dense layer has 22-units, according to the number of attacking classes of the NSL-KDD dataset. Consequently, there are total of 664,982 parameters where 662,422 parameters can be trained.

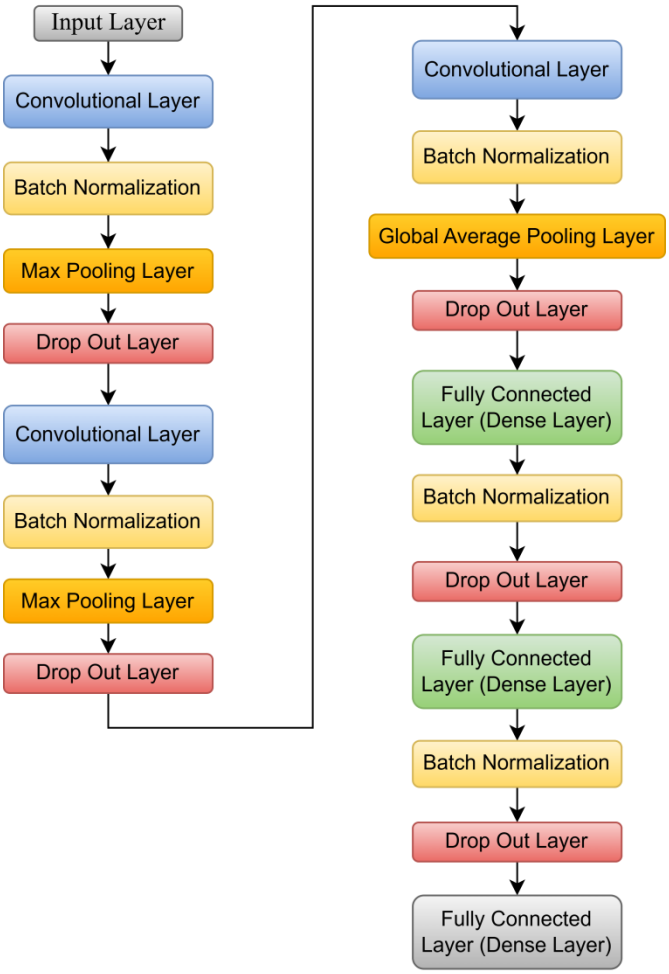

Figure 1. The structure of the proposed DNN classifier.

Simulation Results and Discussion

The suggested method for the dimensionality reduction was discussed and the classifier will be the suggested DNN model in the previous sections. Before simulation, tuning of the FS algorithm and the DNN hyperparameters will be achieved in this section. First, the level of potential solutions examined at each generation is manipulated by the size of the population parameter, which is represented as No. dolphins. The computational cost of a larger population is compensated by a larger number of exploratory capabilities and diversity. Between 8 and 20, values were explored in this study, the best tradeoff between quality of the solution and computing efficiency was discovered with No. dolphins = 15. It was found that with No. dolphins equals to 8, the exploration was faster but not global, when the dolphins become 15, good balance between speed and exploration, while with 20-dolphins, the operation was slower but with improved exploration.

The algorithm makes iterations based on the amount of generations. Convergence may also occur prematurely when there are insufficient generations and underutilization of computing resources when there are too many generations showing no significant advancement; this value dictates the convergence behavior. Generations 1-3 of the algorithm made rapid steps, but generation 4-10 perfected it. This trend was suggestive of first-slow-then-fast convergence. Accordingly, the conducted experiments produced 15, 12, and 9 features, respectively, i.e., with 15-generations, the best reduction (only 9-features) was achieved. On the other hand, the fitness function of the DMA can be described as [16]:

$$Fit = \propto \times accuracy + (1 - \propto) \times \left(1 - \frac{Number\ of\ Features}{Total\ Number\ of\ Features}\right) \quad (7)$$

In this respect, the alpha parameter (∝) is the most significant parameter, which controls the trade-off between reducing features and classification accuracy within the fitness function. When ∝ is close to 1.0 more features are selected which means that the algorithm gives more importance to accuracy rather than feature reduction. At ∝ = 0.5, all features are removed noisily, and this could cause some accuracy to be lost at the expense of a smaller model. Based on this phenomenon, the selected value was ∝= 0.6, where 9-features resulted including: protocol_type, dst_bytes, is_host_login, is_guest_login, count, serror_rate, srv_diff_host_rate, dst_host_srv_serror_rate, and dst_host_srv_rerror_rate.

The sample size parameter can be changed to allow the number of training samples used to be evaluated on the fitness to be altered. This approach is a computational optimization technique that approximates the fitness by a randomly sampled fraction of the dataset (125,973 samples), instead of being applied to all solutions on the entire dataset. This leads to a substantial reduction of time used in calculation without compromising quality of solutions made. The value of 3000-5000 samples was found to give reliable values in terms of fitness. To be more specific, this work adopted 5000-records/samples.

On the other hand, the DNN based CNN model hyperparameters were selected after extensive trails of simulations. Thus, the number of epochs was set to 100-epoch, the optimizer is "adam", the learning rate is 0.001, batch size set to 5600 samples, and the validation split is 0.2, in other words, 20% of training data used for validation-phase. At the end, the tracking metric was the accuracy, as listed in Table 1.

Table 1. Hyperparameters setting of the DNN based CNN model.

Hyperparameter	Seting
Loss Function	categorical_crossentropy
Optimizer	adam
Learning Rate	0.001
Epochs	100
Batch Size	5600
Validation	20%
Metrics Tracked	Accuracy

That is, Figure 2 shows the taining and testing accuracy phases, it can be seen that the model did not over/under fit, since there is no divergence between the two curves which can be confirmed by monitoring the losses of the two curves in Figure 3.

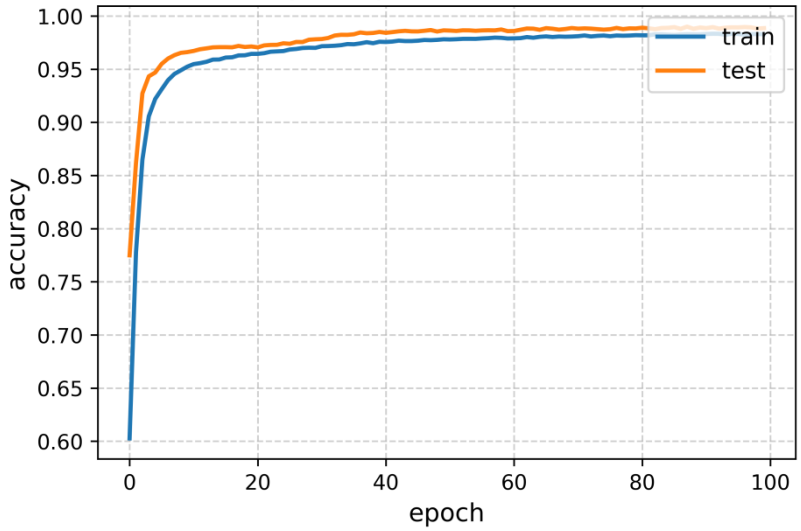

Figure 2. Train and test accuracies of the proposed DNN classifier.

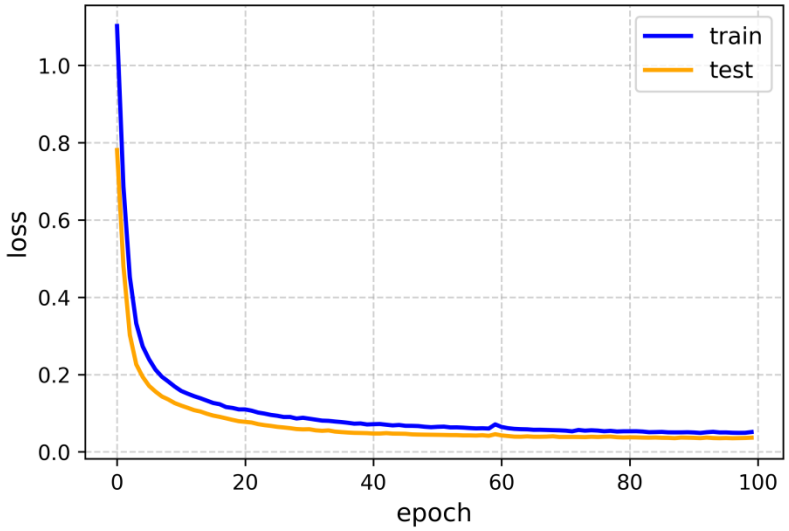

Figure 3. Train and test losses of the proposed DNN classifier.

Consequently, Figure 4 shows the confusion matrix of the test-phase. In such a way, Figure 4 (or Figure 5) shows the confusion matrix of the intrusion detection model proposed on the test set NSL-KDD. The matrix uses a color gradient in the sense that the cells with darker colors have more prediction counts, which makes understanding the classification performance fast. The high score in the diagonal of normal (917 correct) and DoS (2288 correct) is a direct indication of how the model works well in such classes. On the other hand, the lighter diagonal elements of U2R (55) and R2L (113) with significant off-diagonal rating, intuitively accentuate the weakness of the model on minority attack classes. Specifically interesting is the dispersion pattern of the U2R and R2L rows, where wrongly classified instances are spread across several columns - U2R instances are mistaken with DoS (51) and R2L (67) whereas R2L ones are mistaken with normal ones (63), DoS (83), and U2R (18). This visual scheme indicates that the features of different types of attacks are similar, as well as demonstrates that the model is weakest in R2L attacks, as they are the most dispersed. The matrix is therefore a potent diagnostic tool in that it would relay instantly both the strengths of the model and the areas that need to be improved.

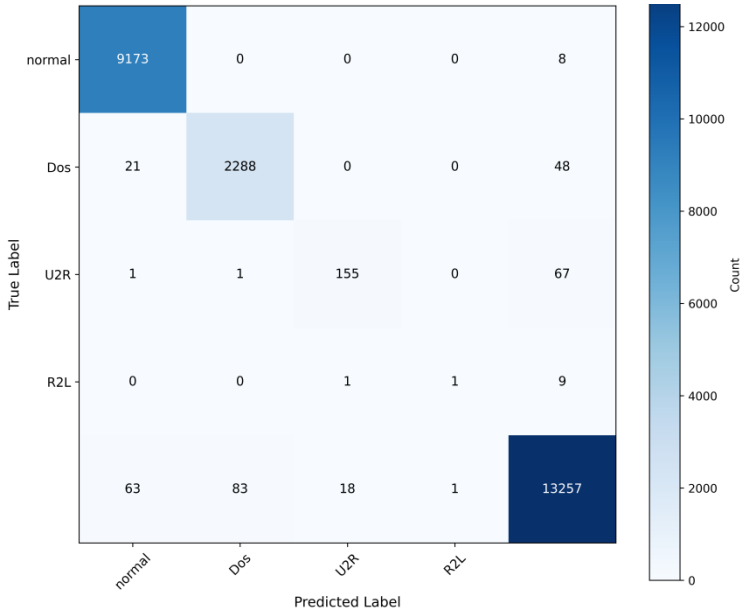

Figure 4. Confusion matrix of the proposed DNN classifier results (counts version).

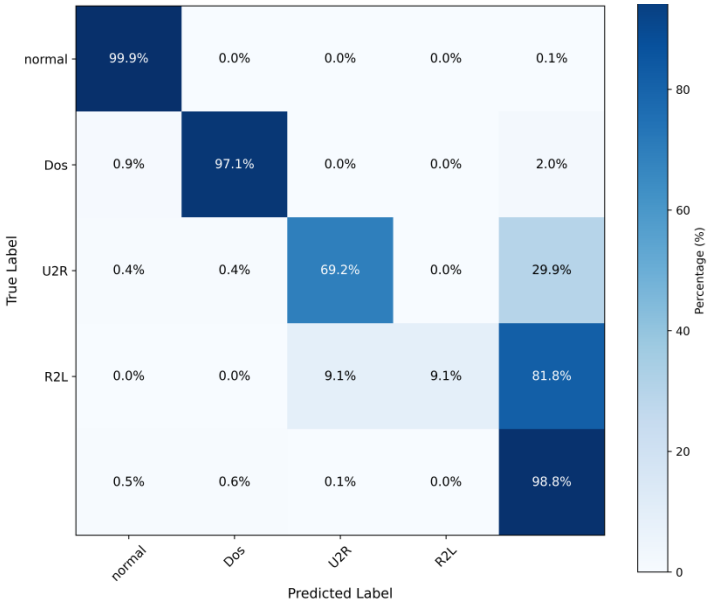

Figure 5. Confusion matrix of the proposed DNN classifier results (percentage version).

The confusion matrix gave a total accuracy of 89.95 percent (3373 correct predictions) with the total of 3750 instances, which is a great result that can be said the proposed model is effective in classifying the majority of the network traffic. This accuracy is high indicating that Dolphin Mating feature selection algorithm is effective in detecting discriminative features in intrusion detection. Normal traffic records very high F1-score of 92.7 which means that the model is very reliable at differentiating between benign and attacks. This criticism of the model is that the slightly greater recall (94.8) than the precision (90.6) indicates that the model is slightly conservative - it sometimes mis-classifies attacks as being normal (false positives), but is almost always false negative (Table 2).

Table 2. Summary of the classification results for the suggested classifier.

Class	Support	Precision	Recall	F1-Score

Normal	967	90.6%	94.8%	92.7%
DoS	2331	93.3%	98.2%	95.7%
U2R	123	75.3%	44.7%	56.1%
R2L	329	62.8%	34.3%	44.4%
Macro Avg	-	80.5%	68.0%	72.2%
Weighted Avg	3750	90.1%	89.95%	91.2%

The best performance of any of the classes is recorded in DoS attacks with impressive recall of 98.2 and the F1-score of 95.7. This means that the chosen characteristics are specifically useful in describing the volumetric properties of DoS attacks that in most cases may have a unique traffic pattern relative to other types of attacks. U2R attacks are a serious issue and only 44.7% recall. This translates to 55.3 percent of U2R attack being undetected, mostly being incorrectly identified as DoS (51 cases) and R2L (67 cases). The fact that the accuracy is slightly higher (75.3%), in cases where the model can actually predict U2R, it is fairly certain, though it cannot distinguish most instances of U2R. Such disparity in performance is indicative of the advanced level of U2R attacks that can include attempts at privilege escalation and can mimic ordinary administration efforts. R2L attacks are the most challenging of the model, and the model has a recall of 34.3% and F1-score of 44.4%. One confusing aspect of the confusion matrix is that R2L attacks are being falsely categorized in many classes: normal (63), DoS (83), and U2R (18) classes. This scattered misclassification behavior indicates that R2L attacks are similar to several other types of traffic and hence they are especially difficult to differentiate. R2L attacks will, in most cases, be unauthorized remote access, and this will have traffic patterns that mimic legitimate remote connections. This difference in the performance between the majority classes (967 samples, DoS 2331 samples) and the minority classes (U2R 123 samples, R2L 329 samples) represents clearly the implication of the class imbalance on the performance of the model. The macro-averaged F1-score of 72.2% is a stark difference to the weighted F1-score of 91.2% as it demonstrates that most of the classes are dominant in the overall measures. This 19-percentage point gap quantitatively proves that class imbalance is a significant source of bias in model performance in favor of frequent classes.

To conclude, the metrics provided by the confusion matrix prove that the proposed model demonstrates high results in normal traffic and DoS attacks (F1-scores >92%), intermediate results in U2R attacks (F1-score 56.1%), and poor results in R2L attacks (F1-score 44.4%). The extreme performance difference between the majority and the minority classes proves that the issue of imbalance between classes is one of the most basic in intrusion detection. The next step is to improve techniques that will enable the minority classes to be better identified, e.g., SMOTE oversampling, cost-sensitive learning, or imbalance-specific ensemble.

Conclusion

The findings of this paper demonstrated that a Dolphin Mating algorithm which bases its observations on the nature can be used to detect network intrusions based on the detection of compact and discriminative features subsets. Arguably, reducing intrusion detection systems by 78 characteristics to 9 characteristics (reduction of 41) is a major move in the right direction. The identified set of characteristics offers valuable data and understandable information about the characteristics of network attacks. The gap between the performance of minorities and majority groups is substantially large which makes it obvious that the research in the field of intrusion detection is still concerned with dealing with the issue of the class imbalance. Nevertheless, additional studies on nature-inspired feature selection, class imbalance reduction, and real-time intrusion detection implementation can be based on the architecture that is developed in this paper. It is essential to develop efficient, interpretable and accurate detection systems and particularly as network threats get more complicated and more recurrent. This study is a step towards a correct

direction as it demonstrates that biological-inspired algorithms can contribute to this significant cybersecurity issue.

REFERENCES

[1] Index, Cisco Visual Networking. "Global mobile data traffic forecast update, 2016–2021 white paper." *Cisco: San Jose, CA, USA* 7, 180, (2017).

[2] M. Baykara and R. Daş, "Saldırı tespit ve engelleme araçlarının incelenmesi," *DÜMF Mühendislik Dergisi*, vol. 10, no. 1, pp. 57–75, Mar. 2019, doi: 10.24012/dumf.449059.

[3] K. He, D. D. Kim, and M. R. Asghar, "Adversarial Machine Learning for Network Intrusion Detection Systems: A Comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 538–566, Jan. 2023, doi: 10.1109/comst.2022.3233793.

[4] B. Deore and S. Bhosale, "Adaptive Dolphin Atom Search Optimization-Based DRNN for Network Intrusion Detection System," *SN Computer Science*, vol. 4, no. 5, Jul. 2023, doi: 10.1007/s42979-023-02006-6.

[5] M. Bakro *et al.*, "Building a Cloud-IDS by hybrid Bio-Inspired feature selection algorithms along with random forest model," *IEEE Access*, vol. 12, pp. 8846–8874, Jan. 2024, doi: 10.1109/access.2024.3353055.

[6] E. H. Salman, M. A. Taher, Y. I. Hammadi, O. A. Mahmood, A. Muthanna, and A. Koucheryavy, "An anomaly intrusion detection for High-Density Internet of Things wireless Communication network based deep learning algorithms," *Sensors*, vol. 23, no. 1, p. 206, Dec. 2022, doi: 10.3390/s23010206.

[7] A. H. A. Hussain *et al.*, "Urban Traffic Flow Estimation System based on Gated Recurrent Unit Deep learning Methodology for Internet of Vehicles," *IEEE Access*, vol. 11, pp. 58516–58531, Jan. 2023, doi: 10.1109/access.2023.3270395.

[8] J. Cheng, Q. Zou, and Y. Zhao, "ECG signal classification based on deep CNN and BiLSTM," *BMC Medical Informatics and Decision Making*, vol. 21, no. 1, p. 365, Dec. 2021, doi: 10.1186/s12911-021-01736-y.

[9] W. Ullah, I. Siddique, R. M. Zulqarnain, M. M. Alam, I. Ahmad, and U. A. Raza, "Classification of arrhythmia in heartbeat Detection using Deep Learning," *Computational Intelligence and Neuroscience*, vol. 2021, no. 1, p. 2195922, Jan. 2021, doi: 10.1155/2021/2195922.

[10] M. Morshed and S. A. Fattah, "A deep neural network for heart valve defect classification from synchronously recorded ECG and PCG," *IEEE Sensors Letters*, vol. 7, no. 9, pp. 1–4, Aug. 2023, doi: 10.1109/lsens.2023.3307053.

[11] Z. Salek, F. M. Madani, and R. Azmi, *Intrusion detection using neuarl networks trained by differential evaluation algorithm*. 2013, pp. 1–6. doi: 10.1109/iscisc.2013.6767341.

[12] E. Popoola and A. O. Adewumi, "Efficient feature selection technique for network intrusion detection system using discrete differential evolution and decision.," *Int. J. Netw. Secur.*, vol. 19, pp. 660–669, Jan. 2017, [Online]. Available: http://ijns.jalaxy.com.tw/contents/ijns-v19-n5/ijns-2017-v19-n5-p660-669.pdf

[13] V. Hajisalem and S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," *Computer Networks*, vol. 136, pp. 37–50, Feb. 2018, doi: 10.1016/j.comnet.2018.02.028.

[14] M. Mazini, B. Shirazi, and I. Mahdavi, "Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms," *Journal of King Saud University - Computer and Information Sciences*, vol. 31, no. 4, pp. 541–553, Mar. 2018, doi: 10.1016/j.jksuci.2018.03.011.

[15] B. A. Tama, M. Comuzzi, and K.-H. Rhee, "TSE-IDS: a Two-Stage classifier ensemble for Intelligent Anomaly-Based Intrusion Detection System," *IEEE Access*, vol. 7, pp. 94497–94507, Jan. 2019, doi: 10.1109/access.2019.2928048.

[16] M. S. Et. Al, "Network based intrusion detection using the UNSW-NB15 dataset," *International Journal of Computing and Digital Systems*, vol. 8, no. 5, pp. 477–487, Jan. 2019, doi: 10.12785/ijcds/080505.

[17] F. Gao and M. Abisado, "Enhanced feature engineering symmetry model based on novel Dolphin swarm algorithm," *Symmetry*, vol. 17, no. 10, p. 1736, Oct. 2025, doi: 10.3390/sym17101736.

[18] J. Rachel and E. Devarasan, "Robust contactless fingerprint authentication using dolphin optimization and SVM hybridization," *Frontiers in Big Data*, vol. 8, p. 1641714, Dec. 2025, doi: 10.3389/fdata.2025.1641714.

[19] A. Kaveh and N. Farhoudi, "A new optimization method: Dolphin echolocation," *Advances in Engineering Software*, vol. 59, pp. 53–70, Apr. 2013, doi: 10.1016/j.advengsoft.2013.03.004.

[20] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, Jan. 2018, doi: 10.1109/tetci.2017.2772792.

[21] T.-Q. Wu, M. Yao, and J.-H. Yang, "Dolphin swarm algorithm," *Frontiers of Information Technology & Electronic Engineering*, vol. 17, no. 8, pp. 717–729, Aug. 2016, doi: 10.1631/fitee.1500287.