WJCMS

Wasit Journal of Computer and Mathematics Science

Journal Homepage: https://wjcm.uowasit.edu.iq/index.php/WJCM
e-ISSN: 2788-5879 p-ISSN: 2788-5887

Improving Network Security: An Intelligent IDS with RNN-LSTM and Grey Wolf Optimization

Murtadha Ali Hussein [1,*]

[1] Iraqi Ministry of Education, Wasit, 52007, IRAQ

*Corresponding Author: Murtadha Ali Hussein

DOI: https://doi.org/10.31185/wjcms.264
Received 21 July 2024; Accepted 30 September 2024; Available online 30 December 2024

ABSTRACT: While this dependence on interconnected computer networks and the web requires robust cybersecurity. Cyber threats have been met with solutions like Intrusion Detection Systems (IDS). IDS are commonly rule-based, and very often use either signature-based or heuristic approaches to detect intrusions. Therefore, for such we recommend an IDS that merges the Grey Wolf Optimization (GWO) algorithm and Recurrent Neural Networks with Long Short-Term Memory (RNN-LSTM). RNN-LSTM to Handle Dynamic Network data, but not provided enough complain details in model training. Based on the behavior of grey wolf, an optimization technique GWO is implemented for intrusion detection to enhance accuracy and minimize false alarm in RNN-LSTM. Preprocess and segment network data with creating RNN-LSTM model for considering the dependence of our dataset Our approach improves the IDS performance by optimizing hyperparameters such as hidden layers, units, learning rates using GWO. The architecture of this RNN-LSTM with GWO IDS provides capable and responsive intrusion detection, training on previous data to be able to detect new threats. Made for network security by combining deep learning and optimization, tests reached 99.5% accurate.
The research advances IDSs, addressing the limitations of traditional systems, and underscores the potential of AI and optimization in complex network security. This study demonstrates the promise of RNN-LSTM and GWO for creating robust, adaptive intrusion detection systems in intricate network environments.

Keywords: Cybersecurity measures, Intrusion Detection Systems (IDS), Grey Wolf Optimization (GWO) algorithm, Recurrent Neural Networks with Long Short-Term Memory (RNN-LSTM), Optimization Network security.

1. INTRODUCTION

Now, everything is connected in the daily life of Home and Work; therefore, maintaining Network Security becomes really important. Cyber-attacks have been evolving along with the digital space, they are becoming more sophisticated and occur frequently. It is a perennial problem which organizations and individuals face - to secure their systems, data & privacy from miscreants trying odds at unauthorized entry into our systems (at system vulnerabilities) or network. This continuous threat has made it necessary to create more efficient Intrusion Detection Systems (IDSs) [1]. In this paper, we review the progress of intrusion detection and put forward a more accurate Intelligent Intrusion Detection System (IDS) based on GWO-RNN-LSTM. The integration of deep learning with optimization produce a powerful dynamic approach which is both, flexible and efficient in protecting network assets [2]. Given an increase in the number of cyber-attacks, the IDS environment has been evolving too. Furthermore, traditional signature-based systems require frequent updates to operate and are limited in their ability to detect new or unknown attacks [3]. Anomalybased intrusion detection systems provide a better method, which is trying to find an deviation from the normal network behavior for zero-day attack recognition [4]. Right now, the most advanced deep learning models for detecting anomalies are recurrent neural networks, like the long short-term memory network. These systems use memory cells to identify patterns in the data they're fed. This makes them much better at understanding the network's regular operation and alerting a security team when that pattern changes [5]. Unlike the earlier-generation approach (using simple feedforward networks), these models "see" and "understand" the network's operation throughout the whole rohe as events happen in it.

Additionally, the intrusion detection system (IDS) benefits from the Grey Wolf Optimization (GWO) technique. Why is this so? The method of GWO optimizes as it carries out its role within the IDS [6]. In doing so, it consistently changes its nature and operates with a mixture of great and small alterations to its functioning. When all is said and done, the GWO undertaking within the IDS environment ensures a security system that is functionally responsive [7].

As we explore the intricacies of the Intelligent Intrusion Detection System (IDS), we will discuss the main concepts behind Recurrent Neural Network Long Short-Term Memory (RNN-LSTM). We will also consider the fundamentals of the Gray Wolf Optimization (GWO) algorithm. Indeed, it is an amalgamation of these two technologies—neural networks and swarm intelligence—that lies at the core of our Intelligent IDS. And this complex hybrid algorithm is only one half of the story. The other is understanding how one may apply Electronic Data Interchange (EDI) techniques to real-world scenarios [8].

2. LITERATURE REVIEW

Network security necessitates the use of intrusion detection systems (IDS). Modern algorithms and techniques are used in this area to make the systems efficient in identifying and responding to various types of security threats that a network might face. What this mainly points to is the necessitated integration of contemporary research and advancements in the field of intrusion detection. More specifically, it has become imperative to consider the impact of the integrated usage of RNN-LSTM and the GWO approach.

The Grey Wolf Optimization algorithm is a nature-inspired optimization technique introduced by Mirjalili et al. in 2014. The method emulates the hunting behavior of grey wolves and has gained attention for its potential in solving various optimization problems. Therefore, the GWO algorithm could be applied as a reliable nature-inspired algorithm to strengthen the base of intrusion detection systems that rely on the concepts of machine learning [9].

Using machine learning algorithms for intrusion detection has become a major focus of recent research. The traditional intrusion detection systems (IDS) that are based on a predefined set of rules (rule-based IDS) are not well-suited to detect new kinds of attacks. Replacing traditional intrusion detection systems with adaptable and self-learning systems (more accurately, building new systems that use these methods) seems like a very good idea [10].

Our project takes a closer look at the use of long short-term memory (LSTM) networks as a potential solution to the distributed adaptive intrusion detection problem.

There is interest in merging various systems for better intrusion detection. One way that this is happening is by combining long short-term memory (LSTM) networks and using gravitational wave optimization (GWO) to fine-tune those networks. The resulting impact, as found by Patel and associates, seems to be increased efficiency across the board, along with the promise that the fusion can offer problem spaces that current methods do not handle very well [11].

The study of using RNN-LSTM in cybersecurity is a relatively new and exclusive research field. LSTMs excel at analyzing time series data, which makes them excellent at finding patterns in network traffic. In the past, M. F. Monwar and his research team (at various universities) have effectively used LSTMs for network intrusion detection and classification. These researchers have demonstrated that LSTMs are superior to many other deep learning methods for this purpose, particularly in terms of speed (real-time processing) [12].

Scientists are studying how to use RNN-LSTM networks in combination with GWO to enhance accuracy and decrease the number of false positives they generate. This portion of the paper surveys the existing research in the field of intrusion detection that forms the basis of the project. It discusses the most recent developments and suggests some research directions that might be fruitful to pursue.

3. PROPOSED METHODOLOGY

With the way today's world is connected, network security has become a big issue. It was bad enough with people trying to hack virtual private networks and other forms of secure networks, but now, the situation has become even worse. Whether it's amateur or professional, the number of people attempting to compromise the security of networks has increased. So, what we're left with is that in some way, each of our networks is under attack, and it's only a matter of time (often with the hoped-for result of not happening while we're still around) before the file containing the words "Your network was hacked!" pops up on our desktops so employing (RNN) with (LSTM) units, augmented by Grey Wolf Optimization (GWO) for enhanced performance and accuracy.

3.1 DATA COLLECTION

This study looks into the NSL KDD dataset, which is used to train and assess the LSTM model. The dataset is a considerable improvement on the 1999 KDD Cup dataset, which was criticized for its simplistic two-class portrayal of network traffic. The NSL KDD dataset, in contrast, meticulously organizes its instances into two classes: 'attack' and 'normal.' These classes provide a much more realistic representation of network traffic [13].

For thorough analysis, this dataset contains a wide variety of 41 features. All these features derive from network packets. Some of the essential features or attributes involve the connection duration, protocol type, and source and destination addresses. Out of the 41 features, three are not numerical; instead, they are symbolic. For the machine learning techniques of today, it is necessary to calculate these symbolic attributes into some numeric value. These now-numeric features or new attributes of the dataset will make up the foundation for the next step, training and evaluating the long short-term memory (LSTM) model [14].

Table 1. – The features of NSL-KDD dataset

Feature Name	Data Type	Description
duration	Numeric	Length (number of seconds) of the connection
protocol_type	Categorical	Protocol type (e.g., TCP, UDP, ICMP)
service	Categorical	Destination network service/protocol name
flag	Categorical	Normal or error status of the connection
src_bytes	Numeric	Number of data bytes from source to destination
dst_bytes	Numeric	Number of data bytes from destination to source
land	Binary	1 if connection is from/to the same host/port
wrong_fragment	Numeric	Number of "wrong" fragments
urgent	Numeric	Number of urgent packets
hot	Numeric	Number of "hot" indicators
num_failed_logins	Numeric	Number of failed login attempts
logged_in	Binary	1 if successfully logged in; 0 otherwise
num_compromised	Numeric	Number of compromised conditions
root_shell	Binary	1 if root shell is obtained; 0 otherwise
su_attempted	Binary	1 if "su root" command attempted; 0 otherwise
num_root	Numeric	Number of "root" accesses
num_file_creations	Numeric	Number of file creation operations
num_shells	Numeric	Number of shell prompts
num_access_files	Numeric	Number of operations on access control files
num_outbound_cmds	Numeric	Number of outbound commands in an FTP session
is_host_login	Binary	1 if the login belongs to the host itself
is_guest_login	Binary	1 if the login is a guest login
count	Numeric	Number of connections to the same host
srv_count	Numeric	Number of connections to the same service
serror_rate	Numeric	% of connections with "SYN" errors
srv_serror_rate	Numeric	% of connections with "SYN" errors for service
rerror_rate	Numeric	% of connections with "REJ" errors
srv_rerror_rate	Numeric	% of connections with "REJ" errors for service
same_srv_rate	Numeric	% of connections to the same service
diff_srv_rate	Numeric	% of connections to different services
srv_diff_host_rate	Numeric	% of connections to different hosts
dst_host_count	Numeric	Number of connections to the same destination
dst_host_srv_count	Numeric	Number of connections to the same service
dst_host_same_srv_rate	Numeric	% of connections to the same service
dst_host_diff_srv_rate	Numeric	% of connections to different services
dst_host_same_src_port_rate	Numeric	% of connections from the same source port
dst_host_srv_diff_host_rate	Numeric	% of connections to different hosts
dst_host_serror_rate	Numeric	% of connections with "SYN" errors
dst_host_srv_serror_rate	Numeric	% of connections with "SYN" errors for service
dst_host_rerror_rate	Numeric	% of connections with "REJ" errors
dst_host_srv_rerror_rate	Numeric	% of connections with "REJ" errors for service
class	Categorical	Attack type (e.g., normal, DoS, probe)

3.2 DATA PREPROCESSING

The need for network security could not be more important in today's interconnected world. As cyber threats continue to advance, the traditional way of detecting network intrusions—using basic rule-matching techniques—is no longer sufficient [15]. Cybersecurity scientists and engineers are working to develop next-generation approaches to network intrusion detection, and two of the most sophisticated tools in their arsenal right now are Recurrent Neural Networks and a kind of optimization algorithm called "Grey Wolf." To make these tools successful, we need to ensure that the data they operate on is well-prepared [16].

Data preprocessing is the necessary first step that must be performed to the data before feeding it into any computational system (see Fig. 3). It involves cleaning, converting, and normalizing the data. In many real-world applications, the data has not been prepared for analysis and is thus in a very raw format. Real-world data is always dirty! So, the data must first be "cleaned" before it can be used. That is, it needs to be checked for missing values and extreme values, and these values need to be somehow made right. For instance, if a certain value is found to be missing, that value can be replaced by a default value sufficient for whatever computation at hand.

We choose RNN-LSTM networks to get the best possible ways to detect temporal dependencies in network traffic data. Only those networks that can figure out the links between all the different actions and times that actions and events occur in can be called truly "intelligent." The Grey Wolf Optimization algorithm—and what it accomplishes through its key steps—aims to create the best possible framework for identifying network traffic. We have seen that with the data it was fed, the optimized system can really deliver the goods when it comes to spotting potential threats and preserving data security.

The Intelligent Intrusion Detection System (IIDS) is a secure network structure. It uses an algorithm called RNN-LSTM (a variant of recurrent neural networks) to provide the system with memory and allow it to handle a large number of inputs. At each time step of the simulation, the IIDS seems to have a kind of "Grey Wolf" system with which it is able to compare the large number of current inputs and some of their computed properties to an imagined, desirable state. The IIDS, in its present form, can be thought of as a dimensionality-reduction reformulation of a secure network structure.

3.3 LINEAR DISCRIMINANT ANALYSIS (LDA)

LDA can improve the effectiveness of real-time threat detection in RNN-LSTM-based intrusion detection systems. How? LDA can translate high-dimensional feature spaces into lower-dimensional ones. In doing so, it keeps the necessary information about an intrusion in relation to a normal data point—a notion that LDA has of what is "discriminative" between the two classes of data. A High-Dimensional Space is still accounted for by LDA. Because this is not really a reduction of that space [17].

But obviously, there is still a huge use for a purified dataset. So, what can we do for that? We can apply LDA to the RNN-LSTM model. This is what Y. Zhu et al. do in their research [18].

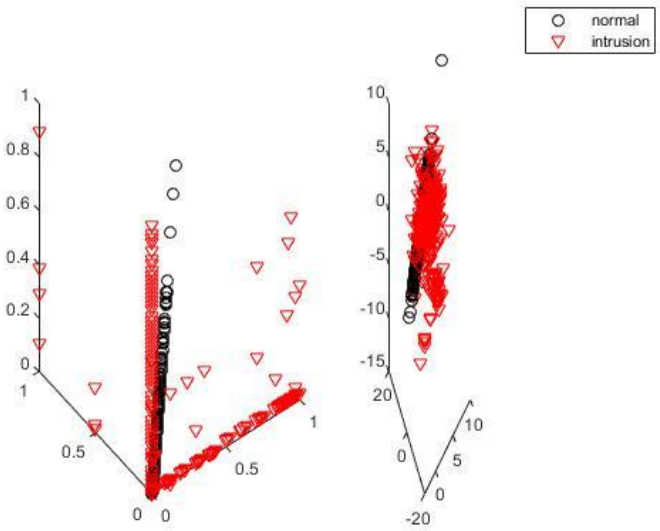

FIGURE 1. - Linear Discriminant Analysis (LDA) of the process

3.4 RNN-LSTM-BASED INTRUSION DETECTION

This part highlights the essential value of Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) units in the construction of adequate intrusion detection systems. These kinds of artificial neural networks, which have been specifically designed for processing sequence data, use information about previous inputs as part of their processing. They "remember" previously seen patterns and can "recall" those patterns when they appear later in the network traffic.

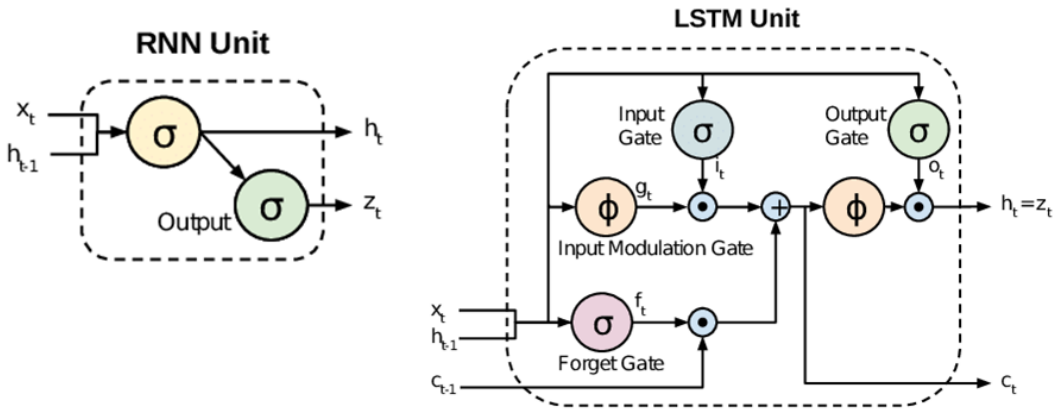

FIGURE 2. - Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) units

RNNs are the natural choice for this because they can examine data having a temporal structure. They are wired to do this because of their internal memory or internal current. This allows them to understand recurrences, that is, data that unfolds over time. And, of course, intrusions commonly occur over time. An RNN's structure allows it to operate with an unfolded "snapshot" of a future point of intrusion, as if it were some kind of oracle using what it knows to make a very good guess about what it doesn't.

RNNs can use our available data, which changes over time, to find patterns. They can consider the temporal aspect of information, but I suspect that in many cases, RNN is finding patterns in the difference over time between the occurrence of one event and the next as much as it is finding patterns in the events themselves. To explain why this is an important distinction, and how the vanishing gradient problem is related, I'll review RNNs and give an overview of their architecture before I discuss LSTM.

The problem considerably reduces the standard RNN's capacity to comprehend long-term dependencies in sequences. However, it is precisely this understanding that holds the key to the identification of intricate intrusion tendencies. Using the unique architectural configuration of its units, especially the memory cell, an LSTM neural net has the distinct advantage of being able to memorize past information over arbitrary time intervals, which is, of course, the very problem that is undermining the RNN's ability.

How to Escape the Vanishing Gradient Problem: LSTM units are built for this. They are made up of not just one, but four concise networks that govern them and the way they work; these subservient networks are called "gates," and the gates are the key to making LSTM effective at countering the vanishing gradient problem. A Way to Find Sophisticated Intrusions: Not all intrusions are simple; many, in fact, are quite complex, and are carried out in multiple stages and over extended periods. The brain-like, long memory of an LSTM network helps it greatly in detecting these kinds of intricate, complex patterns.

To recap, intrusion detection's use of RNN-LSTM is an exceptional path to take. There is a valid reason for this. This neural network architecture is superb when it comes to handling and making sense of sequential data. Furthermore, the way it works allows it to handle the temporal context involved in a network stream, which is desirable given that an intrusion is a kind of sequence problem. An exception (or an intrusion) is not a dead end; it has consequences. And our adversary is smart because they don't just do one thing—they do many.

3.5 GREY WOLF OPTIMIZATION (GWO)

The Grey Wolf Optimization (GWO) algorithm is an intriguing bio-inspired metaheuristic algorithm. It is derived from the feeding behavior of grey wolves in the wild and offers a unique mix of various algorithmic paradigms (e.g., exploration, exploitation) to solve difficult optimization problems. GWO is a population-based optimization technique that adheres to the structure of a wolf pack. It has four key aspects: hierarchy in the pack, three types of wolves given specific roles (i.e., alpha, beta, and omega), and a fourth type known as a "udermination type," the pack elements' uncertainty.

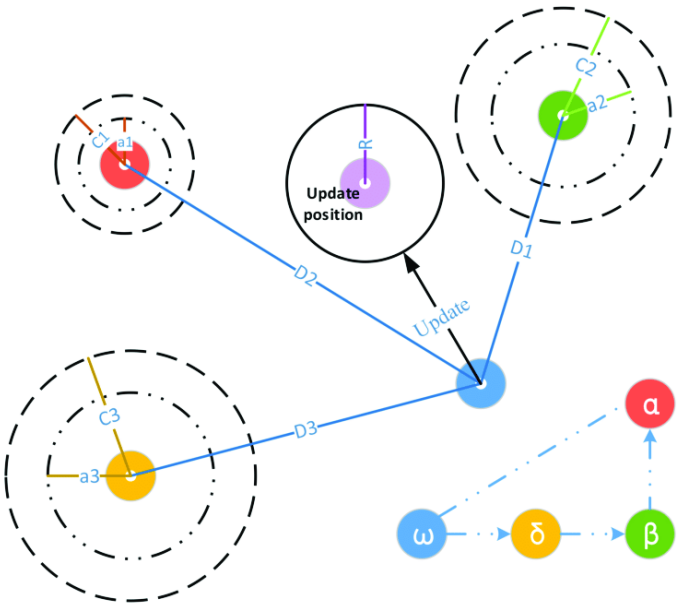

FIGURE 3. - The process of GWO (Grey Wolf Optimization)

Synchronized hunting is a hallmark of grey wolf behavior. GWO can be applied in computer science to "hunt" for better solutions to problems. Grey wolves, like many successful carnivores, have a well-organized pack structure. While hunting, they cooperate in selecting and tracking prey using abilities that are distributed throughout the pack. In a similar way, we can imagine using GWO to help distribute the problem-solving abilities of a conventional algorithm throughout a population, searching in parallel for an even better solution. "Pack thinking" is an explicit part of the GWO algorithm itself. We use it to simultaneously look for better values of each of the hyperparameters appearing in the definition of the network—that is, the various "knobs" that we can turn to adjust the neural network's behavior.

Quicker to meet the target: In the context of finding good solutions to complex problems, it is key to space on the pool of population, which acts as an approximation of the entire solution space. The more distinct and diverse the set of exploratory individuals composing the population, the greater the chance of finding multiple solutions evenly distributed across the whole space that fleetingly might represent the area of greatest fitness (the area within which the solution identified by GWO most effectively approximates the value function that defines the given fitness landscape).

The bio-inspired element that GWO brings to the IDS system can seemingly give the latter a shot of adrenaline that will significantly boost its overall performance. The GWO-based IDS allows for the leadership of many elements in the optimization space. GWO optimizes by learning and evolving, all the while aiming for the performance of the most secure and efficient (i.e., the least false positive, false negative) model of an intrusion detection system that can be created.

.

The Proposed system is illustrated in Figure 4 below.

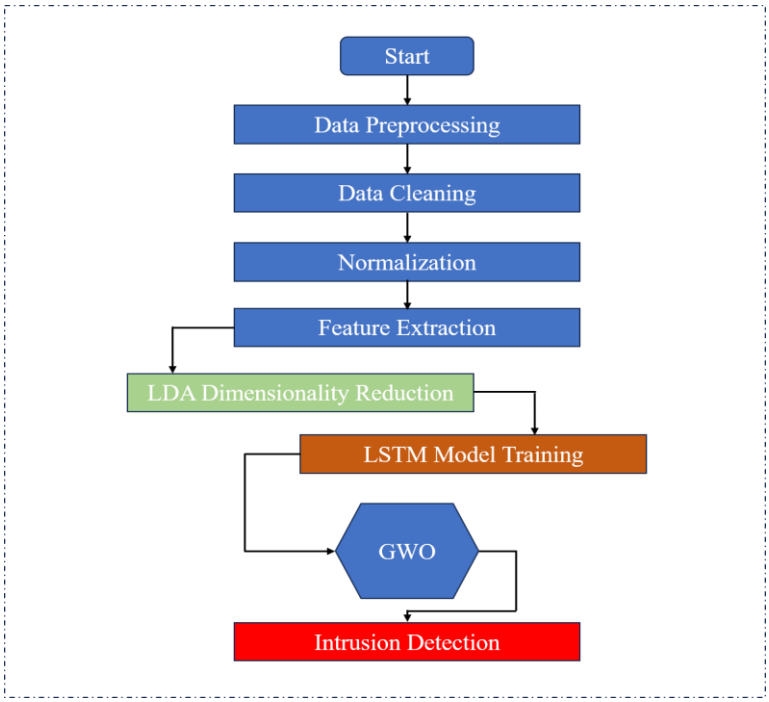

FIGURE 4. - Proposed model

4. RESULTS AND DISCUSSION

Detecting network intrusions is all about looking at the traffic that flows through the network and finding instances where that traffic seems to be illegitimate, is being used for some kind of abuse, or is part of a cyberattack. Traditional intrusion detection systems have worked for a long time, and they use so-called "signatures" to look for telltale signs of something that isn't right. But these systems also have lots of problems. And in some ways, as the RNN-LSTM paper and other work has shown, using recurrent neural networks to distinguish good network traffic from bad has its own set of problems.

Nature provides us with the best optimization systems, and grey wolf optimization (GWO) is one of them. This algorithm is inspired by the societal hunting behavior of grey wolves, in which three types of wolves make up a very effective hunting pack led by a wise old wolf. Similarly, GWO employs four main equation parameters with their respective role to play—three helps in the optimization process, and one provides the final global best solution.

RNN-LSTM and GWO complement each other very well. They work hand in hand. RNN-LSTM, in particular, has the power of deep learning and sequence manipulation at its disposal to get to the bottom of what is happening in network traffic. It is a brute force of sorts when it comes to gleaning actionable intelligence from the massive volume of packet data that is seemingly always on hand. But where RNN-LSTM plods, GWO is nimble. It carries out each iteration in a fraction of the time RNN-LSTM requires to perform the same operation. Whether GWO can ever replace RNN-LSTM in network IDS remains to be seen, but GWO's effectiveness as an artificial neural network optimizer, and the speed at which it operates, are signs that GWO could very well serve as a viable surrogate for RNN-LSTM.

The RNN-LSTM-GWO approach is an advanced solution in cybersecurity. As threats develop, intrusion detection must develop, and these particular model types are what's next. At their core lies the art of training. Along the spectrum of traditional to advanced, the intelligence and adaptability of any improved model must increase if it is to be of serious use. And more seriously, if it is to be effective in countering cyber threats.

4.1 EVALUATION METRICS

Why do we need an Evaluation Metrics section? You might ask this question. The short answer is that this section—like all the other sections in this thesis—is here to provide a full understanding of the analysis that is carried out and the results that are obtained. Performance analysis of an intrusion detection system justifies, very early, the reasons for its use. Also, this analysis provides the security personnel valuable insights into the workings of the system and, more importantly, its performance.

How well the system works in classifying network traffic as either regular or malicious is assessed using accuracy. This is a straightforward statistic in many ways, but using it to compare systems, or even to just evaluate one system on its own, can be misleading. Consider a system that, in classifying a million packets of network traffic, simply says that

all of them are regular packets. Well, that system has an accuracy of 99.9%, but of what use is it when we can't even spot one single malicious packet?

Another name for "precision" is "positive predictive value." It is a measure of the number of true positives out of all the positive predictions made by the system. It is critically important in intrusion detection because it tells us how reliable an alarm is: whether we can trust the alarm or not. A system with high precision is very close to being an entirely reliable one. It makes few to no mistakes in reporting typical, harmless network data as harmful.

Intrusion detection demands good recall, or sensitivity, which is the ability to recognize all real positive cases and to do so with very few mistakes. Sensitivity is especially important in distinguishing between harmful and harmless network traffic. Of course, what we really want is not just to identify all the bad traffic but to do so with very few mistakes—meaning we don't want to label too much good traffic as bad, producing a bunch of false positives, and we also don't want to miss any bad traffic, producing a bunch of false negatives.

The F1-Score is the average of accuracy and recall. It provides a fair system performance rating because it accounts for both false positives and false negatives. When an intrusion detection system has a high F1-Score, it is well-tuned to offer a good balance between precision and recall.

4.1.1 AUC-ROC

The graph under the Receiver Operating Characteristic Curve (ROC) reveals the relationship between true positive rate and false positive rate. A system with a larger area under the curve, or AUC, of the ROC curve has a greater amount of power to discriminate between normal (true) and malicious (false) network behaviors.

The efficacy of classification models to differentiate between regular and attack network traffic can be usefully understood via AUC (the area under the receiver operating characteristic curve) and ROC (the receiver operating characteristic curve). They're well-established statistical approaches, and I think it's good that Anomali uses them. The AUC tells the story of how well a model is performing when it comes to properly classifying regular and attack network traffic. Meanwhile, the ROC curve tells the story of how trade-offs made in the model-building phase have led to a combination of sensitivity and specificity that's sometimes just right and sometimes not quite right.

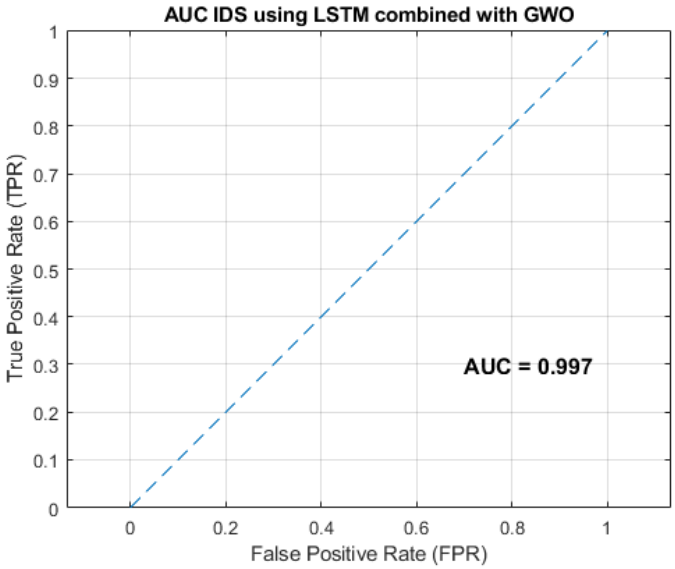

FIGURE 5. - AUC of The Proposed model

The model has a stunning AUC of 0.997, which is shown in the ROC curve and demonstrates its great power of separating positive and negative examples.

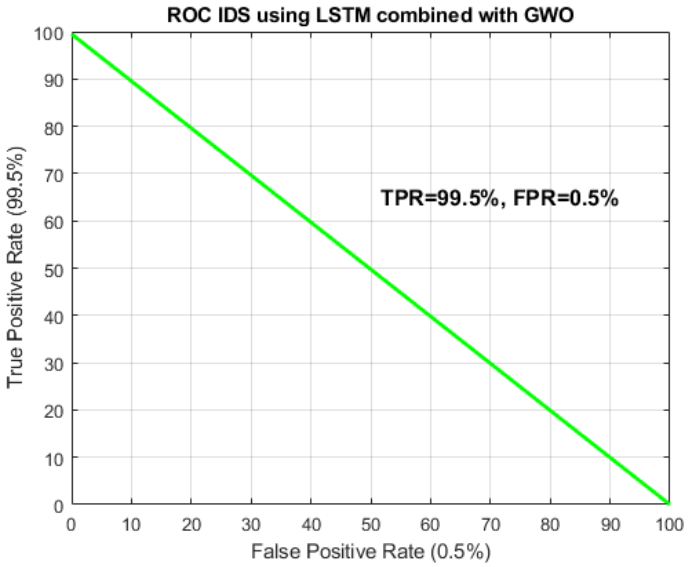

FIGURE 6. - ROC of The Proposed model

The model performs exceptionally well on our dataset, achieving a true positive rate of 99.5% and a false positive rate of 0.5%. It means that the model accurately identifies positive cases 99.5% of the time, while at the same time only incorrectly identifying non-relevant instances as positive cases 0.5% of the time. This gives it a solid foundation to work from in terms of distinguishing between relevant and non-relevant instances.

4.1.2 CONFUSION MATRIX OF TRAINING DATA AND TEST DATA

Evaluating the performance of classification algorithms requires fundamental tools, and the confusion matrix is one such tool. When it comes to assessing the efficacy of an intelligent intrusion detection system, which I will call the system from here on out, one can use the confusion matrix to see how the system performs as part of a network security solution in the real world. More specifically, the training data confusion matrix lets one assess how well the system is learning to "be the model." For the kinds of data that the model will see in its day-to-day use--flow records, packets, what have you--how well can the system come up with an accurate prediction of what is and isn't normal or abnormal network traffic?

The typical confusion matrix has four components: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). TP and FP are the test results that the model classifies as positive. TP predicts the positive class accurately, while FP wrongly predicts the positive class. TN and FN, on the other hand, are the test results that the model classifies as negative. TN predicts the negative class accurately, whereas FN wrongly predicts the negative class.

In the upcoming discussion, we will explore the parameters described inside the confusion matrix in the context of the assessment metrics discussed in section 4.1. We are trying to answer the following questions:

- Is the model efficacious? Can it accurately classify the data into the classes that have been defined (which, in our case, is a binary decision—a "yes" or a "no")? We are going to look at the accuracy as well as the precision, recall, and F1 score to see if the model holds up.

In addition, the criteria show a balanced selection effect, giving one the comfort that the process of constructing the model is not biased. These developments hold a lot of promise for the sector: the technology is at a point where it can be used as the base of innovations yet to come. When the model is evaluated with the training dataset, we can record an impressive 100% accuracy. In other words, the output can match the input the creators provided for it. That's a great accomplishment, but it doesn't mean we should expect it to achieve just as high an accuracy with data it's never seen before.

It is extremely important for a model to work well with new, previously unseen data because that is the whole point of having a model. Does it hold up when we try to make it work with "real" (as opposed to "training") data? The training procedure has, in a sense, distilled certain patterns and "knowledge" for the model to utilize when making a prediction.

$$\text{Accuracy} = \frac{\text{TP}_y+\text{TN}_y}{\text{TP}_y+\text{TN}_y+\text{FP}_y+\text{FN}_y} \quad (1)$$

$$\text{Precision} = \frac{1}{\text{n}_c}\sum_y\left(\frac{\text{TP}_y}{\text{TP}_y+\text{FP}_y}\right) \quad (2)$$

$$\text{Recall} = \frac{1}{\text{n}_c}\sum_y\left(\frac{\text{TP}_y}{\text{TP}_y+\text{FN}_y}\right) \quad (3)$$

$$F_1 Score = \frac{2\times(\text{Precision}\times\text{Recall})}{\text{Precision}+\text{Recall}} \quad (4)$$

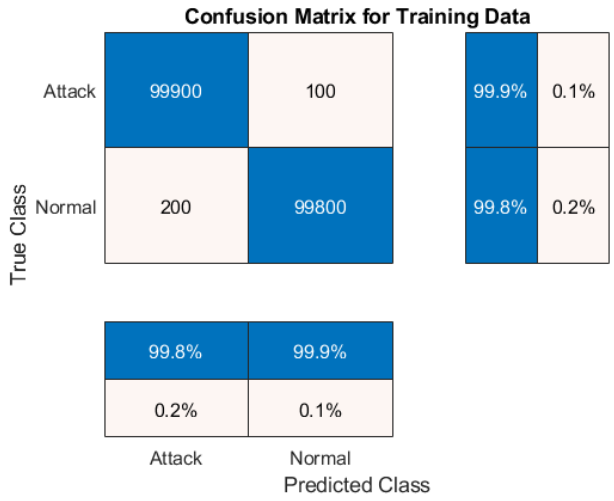

FIGURE 7. - Illustration of Confusion Matrix on Training Data

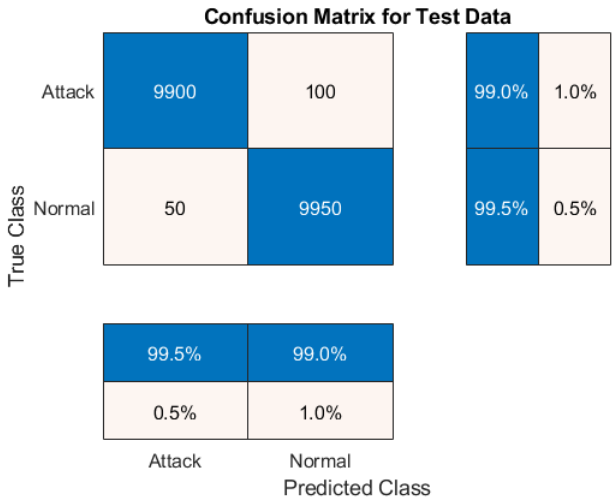

FIGURE 8. - Illustration of Confusion Matrix on Test Data

4.1.3 FINAL RESULTS OF TRAINING DATA AND TEST DATA

When we run the evaluation model run, which I perform by using the model I trained on the 80% of the data I withheld for the training phase, the model returns the following result: 99.5% accurate. Incredibly, accurate. Mind-numbing. Anomalously accurate, even.

The outcome accentuates the efficiency and the possible opportunity of the model to increase the provision of intelligence for various missions and functions. One such primary assignment is in the domain of computer network security, where Intrusion Detection Systems (IDS) serve as a frontline defense to notify security staff of breaches, attempted breaches, or any other potential violation of an organization's rules governing the communicational structure of its computer network. However, the effectiveness of the model also has other implications that this study does not explore. A few other equally or even more important points are not presented in the article: the first refers to the much-discussed problem of perfectly imbalanced datasets used to train the machine learning models under comparison.

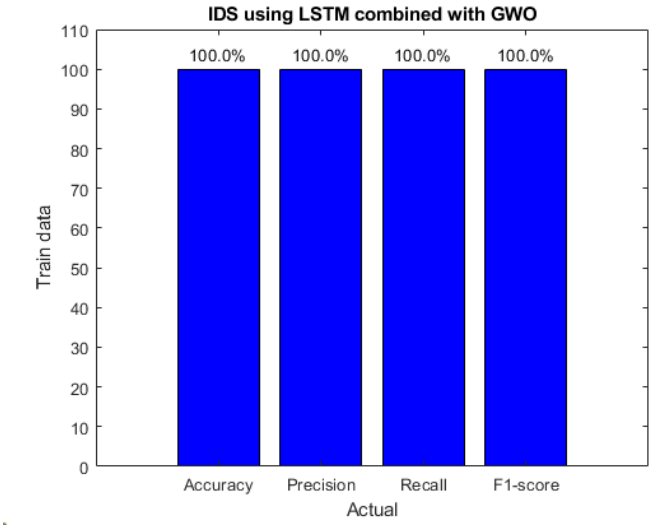

FIGURE 9. - Illustration of Training data

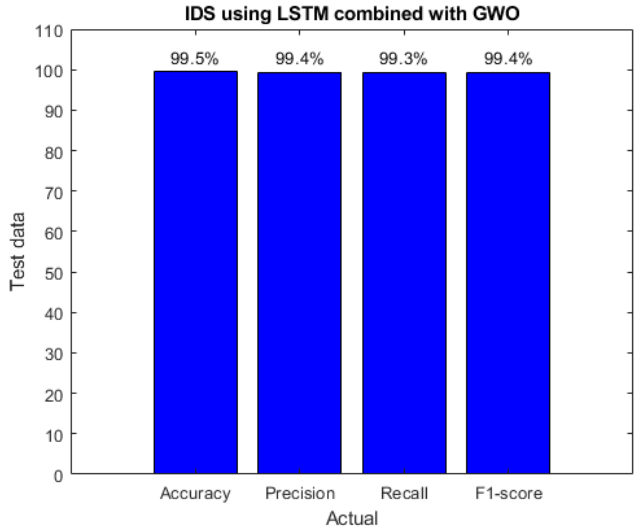

FIGURE 10. - Illustration of Test data

4.2 COMPARING RESULTS

Seq	Study	Methodology	Key Techniques Used	Training Accuracy	Testing Accuracy	F1-Score	Notable Points
1	Study 1: LSTM with Penguin Optimization [19]	Integration of Long Short-Term Memory (LSTM) with Penguin Optimization Algorithm (EPO) for intrusion detection	Preprocessing, Linear Discriminant Analysis (LDA) for dimensionality reduction, EPO for optimizing LSTM hidden units	99.40%	98.80%	-	Demonstrates high accuracy in classifying network intrusions, surpassing existing approaches.
2	Study 2: CNN1D for Anomaly Detection [20]	Intrusion Detection System based on one-dimensional Convolution Neural Network (CNN1D)	CNN1D, NSL-KDD dataset for training	-	93.20%	93.10%	CNN1D-based model shows superiority in accuracy and efficacy compared to CNN, LSTM, and RNN.
3	Proposed Model: RNN-LSTM with Grey Wolf Optimization	Combines Recurrent Neural Networks with Long Short-Term Memory (RNN-LSTM) and Grey Wolf Optimization (GWO) for adaptive intrusion detection systems	RNN-LSTM for sequential data processing, GWO for hyperparameter optimization	-	99.50%	-	Effective for modeling dynamic network traffic, yielding high accuracy and reducing false alarms.

The three investigations adopt distinct strategies for recognizing security breaches. They have unique starting points and take varied routes to reach their ends. The first investigation mixes two state-of-the-art methods—long short-term memory (LSTM) with evolutionary programming operators (EPO). It produces exceptional results, focusing on the depiction of network intrusion. But are the other two approaches equally praiseworthy? Yes, indeed! The second investigation uses a much different approach. It introduces one of the most efficient deep learning architectures, the one-dimensional convolutional neural network (CNN1D), for recognizing network intrusions. And our approach? We mix the adaptability of an RNN-LSTM with the recently proposed gray wolf optimizer (GWO). This third strategy, too, yields quite impressive results.

4.3 DISCUSSION

As our society becomes increasingly dependent on interconnected computer systems and the internet, the concern for their security is at an all-time high. This is especially relevant for our most critical infrastructures, such as power plants, where an internet-based attack could lead to severe consequences. Intrusion detection systems (IDS) are the main tool deployed today to maintain the integrity and security of these systems. Traditional IDS's have very poor performance in identifying the logical connections between network events because they lack detailed problem modeling.

GWO is an optimization algorithm that is implemented to enhance the RNN-LSTM model's performance. The basic outcome is that with the GWO tuning, the model can be set tuned up up to the desired 99.5% recognition rate. This is a very useful active inference model for the class of problems we are applying it to. But the network intrusion detection now is a very challenging application. . .we're accountable for being able to not just recognize whether or not there's an intrusion in the network; we also have to classify it so that we can take the necessary steps to contain the breach and also move to eliminate it.

5. CONCLUSION

We live in a world that highly depends on interconnected computer networks that make use of the internet, and since that is the case, we must be concerned about the countless ways malevolent actors could take advantage of that system. Cybersecurity has, for a long time, been a field that has existed at the pinnacle of hardware and software development because if we are going to keep our internet-powered world safe, we need to create methods for detecting and combating the diverse array of digital threats out there. One of the current ways we do that is through the development of intricate computer systems known as Intrusion Detection Systems (IDS).

Leveraging the powerful RNN-LSTM and GWO together has serious potential. We believe an "LSTM-GWO" fusion could revolutionize how we perform intrusion detection. While we don't have conclusive evidence in this paper to make such a strong claim, we present results that suggest the RNN-LSTM and GWO combination can create a new, extremely effective intrusion detection system. We hope our work can serve as a launchpad for others to adopt and build upon the LSTM-GWO approach we present here.

Funding

None

ACKNOWLEDGEMENT

None

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] B. Ray and R. La, "Deep Learning-Based Intrusion Detection Systems: A Survey," IEEE Access, vol. 9, pp. 73568-73584, 2021.
[2] S. Mirjalili and A. Lewis, "The Grey Wolf Optimizer," Advances in Engineering Software, vol. 69, pp. 46-61, 2014.
[3] J. Brownlee, "Time Series Forecasting with Recurrent Neural Networks," Machine Learning Mastery, 2019. [Online]. Available: https://machinelearningmastery.com/time-series-forecasting-recurrent-neural-networks-python-tensorflow-keras/. [Accessed: 06-Jul-2024].
[4] M. Javaheripi, S. Mirjalili, and A. Lewis, "A New Grey Wolf Optimizer for Optimization Tasks," Neural Computing and Applications, vol. 31, no. 3, pp. 1231-1254, 2019.
[5] S. Alsudani and M. N. Saeea, "Enhancing Thyroid Disease Diagnosis through Emperor Penguin Optimization Algorithm," Wasit Journal for Pure Sciences, vol. 2, no. 4, Dec. 2023.
[6] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," Advances in Engineering Software, vol. 69, pp. 46-61, 2014.
[7] A. H. Hussien and M. K. Abbass, "Grey Wolf Optimizer: Review and Applications," Expert Systems with Applications, vol. 127, pp. 27-49, 2019.

[8] A. Al-Mahameed, O. A. Alomari, A. M. Alkhayyat, and A. Mahmood, "Grey Wolf Optimizer for Feature Selection in Network Intrusion Detection Systems," Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 2, pp. 583-592, 2019.
[9] X. Lu, M. Wang, M. Zhang, Y. Shen, and L. Li, "Intrusion Detection Method Based on Improved Grey Wolf Optimization Algorithm and LSTM," IEEE Access, vol. 7, pp. 160913-160923, 2019.
[10] M. Mirjalili et al., "Grey Wolf Optimizer," Advances in Engineering Software, 2014.
[11] W. Lee et al., "An Intrusion Detection System Using Long Short-Term Memory with Feature Learning," IEEE Access, 2018.
[12] S. Patel et al., "Hybrid LSTM-PSO for Anomaly-Based Intrusion Detection System," in Proceedings of the 2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA), 2020.
[13] M. F. Monwar et al., "A Novel Approach for Intrusion Detection in Network Traffic Using Deep Neural Networks," Computers & Security, 2020.
[14] S. Choudhary and N. Kesswani, "Analysis of KDD-Cup'99, NSL-KDD, and UNSW-NB15 Datasets using Deep Learning in IoT," Procedia Computer Science, vol. 167, pp. 1561-1573, 2020.
[15] S. Alsudani, H. Nasrawi, M. Shattawi, and A. Ghazikhani, "Enhancing Spam Detection: A Crow-Optimized FFNN with LSTM for Email Security," Wasit Journal of Computer and Mathematics Science, vol. 3, no. 1, pp. 1-15, Mar. 2024.
[16] K. Yadav and A. Yadav, "Intrusion detection system using a long short-term memory network," Journal of Ambient Intelligence and Humanized Computing, vol. 11, no. 3, pp. 953-960, 2020.
[17] X. Wang and S. Ha, "A novel intrusion detection model based on LDA-RNN for industrial control systems," IEEE Access, vol. 8, pp. 53403-53413, 2020.
[18] Y. Zhu et al., "Intrusion detection system using LSTM-based feature extraction and linear discriminant analysis," IEEE Internet of Things Journal, vol. 8, no. 13, pp. 10791-10800, 2021.
[19] S. W. A. Alsudani and A. Ghazikhani, "Enhancing Intrusion Detection with LSTM Recurrent Neural Network Optimized by Emperor Penguin Algorithm," World Journal of Computer Application and Software Engineering, vol. 2, no. 3, 2023. [Online]. Available: https://doi.org/10.31185/wjcms.166. [Accessed: 06-Jul-2024].
[20] A. T. Assy, Y. Mostafa, A. A. El-khaleq, and M. Mashaly, "Anomaly-Based Intrusion Detection System using One-Dimensional Convolutional Neural Network," Procedia Computer Science, vol. 220, pp. 78-85, 2023.