Integration of Convolutional Neural Networks and Grey Wolf Optimization for Advanced Cybersecurity in IoT Systems

Israa Ahmed Jaddoa [1*]
[1] Department of Information Technologies, Altinbaş University, Türkiye, Istanbul
Email: [1] esraa.ahmed@uoa.edu.iq
*Corresponding Author

***Abstract*—The rapid integration and application of the Internet of Things in daily life have significantly improved connectivity and intelligent control to various devices. However, it has exposed such systems to increased susceptibility to cyber challenges, such as infiltration, data sovereignty, and cyber-attacks. There is a need for an efficient and secure solution to these apparent security concerns which require complex social structures to adapt to various learning lessons quickly. The purpose of this study is to provide an inventive evolutionary operation to enhance the security of IoT networks and by integrating Convolutional Neural Networks and items of Grey Wolf Optimization algorithms – Standard GWO, Modified GWO and Advanced modified GWO. The GWOs were used to include surveillance accuracy layout, hence boosting detection accuracy. The action Lloyd testing found that smaller OWG intelligence (which achieved initially) unlimited interpretations which increased the percentage and was 97.4 %. This approach was further increased with FGWE, achieving 97.7 percentage, and 97.8 2.02% errors. The performance of both was 98.4 and 97.5 for the two classes, respectively. The current study's results reveal the effectiveness of computational development to enhancing secure IoT networks and offer a secure prototype for potential study to optimize the security structure. effet for keynote curricular scenarios due to the system cause and trusty security solutions.**

Keywords*—IoT Security; Convolutional Neural Networks; Intrusion Detection; Adaptive Algorithms; Cybersecurity; Network Security; Adaptive Algorithms; Grey Wolf Optimization; Cyber Threat Detection.

I. INTRODUCTION

Artificial Intelligence is an essential component in the field of information security which makes it easier for identifying the cyber threats to the personnel. Deep learning algorithms, one of the AI techniques, may process several million events, providing a quick response to remote threats altogether. These systems, referred to as AI, are excellent at pattern recognition, "throwing away" e-malware, and training to identify the behaviors associated with ransomware. There are three types of malware, which is a swift technology with plans to disrupt, obtusely steal your information, and excepts for publishing side comments; they do it for money.

However, in the AV-Test Threat Report, concerning this software, the numbers are getting worse and worse as the amount of malware program that occurs daily shines on prominent numbers with more than 35 million new malware being detected daily.

In the past year, the total number of malware samples circulating has risen to 114 million, with January 2021 alone seeing over 607 million malware programs [1]. Hackers continually employ cunning techniques such as polymorphism [2], code obfuscations, and metamorphism [3], making it challenging for traditional anti-malware solutions to keep pace. Significant malware types include backdoors, password stealers, scanners, miners, DDoS attacks, and ransomware, among others [4].

Malware detection methodologies can be categorized into two groups: static and dynamic systems. Static systems employ symbolic execution to dissect code into components like opcode sequences, strings, function call graphs, and API sequences. In contrast, dynamic or behavioral analysis systems typically operate within a sandbox environment, aiming to detect malicious activities by emulating network behavior or capturing system calls. While effective at identifying covert behaviors, these systems are resource-intensive and slow to react to targeted attacks on critical systems. They often fail to detect malware activity in controlled environments due to the limited number of attacks [5][6].

Given these limitations, this paper proposes a new method for enhancing IoT network security through the combination of CNNs with a series of Grey Wolf Optimization algorithms, namely Standard GWO, Modified GWO and Advanced Modified GWO. Presented algorithms were specifically developed to optimize network hyperparameters and increase the precision of intrusion detection while minimizing false positives under dynamic IoT conditions. Each of the algorithms was tested for effectiveness, and the results were presented in varying stages of their development, as set out in the Methodology section [7].

The growing insights of deep learning are increasingly applied across various domains, including fraud detection, malware processing, and image analysis, aiming to master complex patterns across multiple levels [8]-[13]. The introduction of Convolutional Neural Networks (CNNs), which have significantly contributed to advancements in deep learning, marked a new era of data training with multi-layer architectures and millions of parameters, handling datasets of considerable complexity. CNNs have proven essential in the classification of time-series data and as a subfield in malware detection [14]-[15].

II. Related Work

This section will examine the research that relate to the Internet of Things (IoT) networking security in the context of Intrusions Detection (ID). It is mentioned as a factor according to performance outcomes as well as provides positive and negative aspects of the present system.

Da Costa et al. [16] delivered an elaborate article aimed at bringing out an overview of different machine learning techniques implemented with the purpose of identifying IoT network intrusions. This review mainly emphasizes the machine-learning minority corresponds with the evolutionary- based processes. The primary objective of this work is to make the audience aware of the existing literature in this field of research and adds new material for academics who would want to learn about this security issue related to Internet of Things. The work is innovative and includes a lot of different security techniques that are used to protect computer networks and monitor intrusions in digital environment. Although these strategies are aimed at to limit several false alarms, they are, however, still being thought of as a big obstacle that needs to be taken care of in all previous research studies.

In their article, Islam et al [17] aimed to distill the identified IoT profess by using a set of conventional and deep learning algorithms like SVM, DT, RF, LSTM, and DBN. Data analytics ways of doing things are put to use since they are both speedier than the other methods and more efficient in the actions that has not yet been discovered to come from acknowledged attacks. In essence, a primary objective of those framework's lie in an intelligent, safe, and dependable IoT system which can detect its vulnerabilities, act as a strong wall during big scale cyber-attacks and perform a self-recovery.

This work provides a learning approach capable of detecting and enforcing security enforcement rules of not just a general case but also most of the very specific cases. Nimbalkar et al. [18] have made a close examination of the different feature selection techniques for the purpose of deriving the most effective intrusion detection system (IDS) against the threats in IoT networks. By the Information Gain (IG) and Criterion based on Ratio Margin (GR) selection models 50% of the features is reduced which is then provided to the JRip classifier for a proper detection. In addition, the authors aim to have a summary dataset after the operations of data preprocessing which include visualization, selection, cleaning and transformation. Furthermore, (accuracy of suggested classifier validates with and without IG and GR feature selection techniques is assessed. These techniques denoted in this study have proved effective in enhancing the classifiers performance.

In their article, Hindy et al. [19] applied a thorough case study about intrusion detection and classification to evaluate the widest variety of machine learning techniques. For this task, the MQTT-IoT-IDS 2020 dataset must be used as the primary resource, which contains both the unidirectional and bidirectional attributes required for building a classifier with higher efficiency. The use of data-driven method in the paper by Alsaedi et. al. [20] aims at the detection of intrusions affecting both IoT and IIoT network using the telemetry ToN-IoT dataset. The research direction aims at developing a new dataset for this case study which shall help with the overall preparation to large-scale networks for resilience and protection from the network recent cyber threats. Along with that, it analyzed some well-known and forthcoming datasets collectively covering this area of concern having been listed in Table I.

TABLE I. Overview of IoT Security Datasets

Dataset	Heterogeneity of IoT Data Sources	Different Attack Scenarios	Number of Features	Number of Instances	Scenario
AWID	No	Yes	155	458,691	6
ISCX	No	Yes	45	15,570	5
NSL-KDD	No	No	42	125,973	2
UNSW-IoT trace	No	No	8	1,000,000	7
T-IIoT	Yes	Yes	52	50,000	9
BoT-IoT	No	Yes	116	56,800	8
LWSNDR	No	No	29	48,000	4
KDD'99	No	No	41	494,021	1
UNSW-NB 15	No	Yes	49	2,540,000	3

Zhou et al. [21] put forward GNN based IoT security surveillance scheme. Same here, the hierarchical adversarial attack (HAA) generation algorithm was created to detect unknown attacks in a more precise manner. Moreover, along with RWR, a technique can be used which is helpful in assessing the vulnerability of the network by calculating the priority rating of the nodes. This work builds on the UNSW-SOSR 2019 dataset available for free using the suggested model that will undergo comparison with most of the referenced model. On the other hand, this model is not so time efficient as the solution is predicted, it can be the main flaw of this work. Wahab, et al. [22] suggested utilizing a deep learning online technique in the detection of interplanetary networks intrusion. For the purpose of intrusion data stream detection, the data drift detection technique has been applied. The algorithm can detect a deviation in the characteristics. The suggested method involves sustainable strategy of providing dependable intrusion solution. An advantage of this mechanism is it provides support for both the previous standpoint drifting detection and the new one intrusion identification and recognition. Nonetheless, the presented deep neural network method faces high time and computation complexity.

The authors [23] had an adaptive PSO stategeman incorporated into CNN algorithm to identify network intrusions from IoT devices. Surprisingly, the model targets a multi-type of intrusion detection architecture with a predictive probability and reliability level enhanced from current sys- tems. Beyond that, the highest probability is figured out by using the cross-entropy loss function. We are about to show you the performance of the proposed APSO-CNN intrusion detection system which is assessed by the training loss and accuracy measures.

In addition to this, berádalgawad et al. [24] have utilised bi-directional generations adversarial net- work (Bi-GAN) for the recogntiñ of cyber incidents from the IoT 23 dataset. In the following study, the adversarial autoencoder has been utilized with its iterative counterpart, the Bi-GAN model, for identifying a multitude of attacks from the network. Furthermore, 10-fold cross-validation technique is incorporated in proving the adeptness and validation of the described procedure. In line with these findings, it can be inferred that the deep learning techniques used generationally will provide high accuracy levels and a good detection standard.

Kumar created one of the specialized distributed intrusion detection systems for spotting DDoS attacks from the IoT networks [25]. In most cases, an IoT-based security mechanism needs to manage huge amounts of data which are generated from connected IoT devices in a distributed manner and apply appropriate algorithms applicable in feasible architecture. Shukla, et al. [26] demonstrated an artificial neural network (NN) intrusion detection method for IoT security. The poster presented a few machine learning methods utilizing as inside the IoT environment. Another prominent element underscored in the paper is how essential it is to choose the correct data for a model [27]. Discussions are made on the naive Bayesian network intrusion technique and its IDS as well. In essence, it stressed out the importance of the IoT event preparation process being properly classified as well as the consequences of using a multi-hidden naive Bayes multi-classifier model instead of the standard classifier model which found it more effective. On this basis, the OAI position voids the information [28] pertaining to the given node of a single IoT network, ruling that the node is only responsible for that particular node. Therefore, through scale this method can be augmented with the increasing number of sessions to the number of nodes under analysis. In [29] the authors propose a hierarchical fog computing-based hierarchy for Industry 5.0's smart energy-supplying systems. It processes data-heavy analysis from the IIoT devices faster than classical cloud computing does, so adding ABE for security prevents any data leakages.

Albasheer et al. [30] investigated network intrusion detection systems (NIDS) and the issues they are facing, for example, the number of false positives, different strategies of alert correlation, and the way NIDS have impact on network security. An AI-based, cross-platform VPN system for identifying and classifying attack risks developed by [31] researchers. The were proved the effectiveness of an AI extended gradient boosting (XgBoost) algorithm in cyberattack prevention and developing an AI system which could work together with a Cassandra big data system. In [32], researchers have suggested an on-field intrusion detection system (OMIDS) for electric vehicle networks which show high exactness for different threats. In [32], the authors discovered that supervised machine learning could be applied to IDS with a high accuracy. In [33], the authors raised "IntruDTree" as a machine learning based intrusion detection system created for the IoT.

The literature review has proposed (Table II) that some strategies can be used to decrease the false positive rate, but at the same time, adding more training and labeling is needed. Nevertheless, there are some methods which are in the same direction and on the contrary of this reverse the process which results in reduction in false positives but at the cost of high computational costs for both training and testing. However, this is yet another important challenge for intrusion detection as real-time detection is quite relevant.

TABLE II. SUMMARY OF STATE-OF-THE-ART STUDIES FROM LITERATURE REVIEW

Ref.	Key Contributions	Limitations	Approach
[19] Hindy et al.	Focus on MQTT-IoT-IDS 2020 dataset and bi-directional features	No explicit mention of Limitations	Case study on machine learning techniques for intrusion detection
[24] Abdalgawad et al.	Effective use of Bi-GAN and generative deep learning	No specific limitations provided	Bi-GAN model for detecting cyberattacks in IoT networks
[22] Wahab et al.	Detection of intrusion data streams with drift detection	High computational and time complexity	Online deep learning approach with data drift detection
[17] Islam et al.	Focus on quick implementation and effective handling of un-known events	Limited discussion on computational efficiency	Use of machine learning and deep learning algorithms for IoT threat detection
[16] Da Costa et al.	Comprehensive review of IoT intrusion detection techniques	High false positive rate across research studies	Review of machine learning techniques for IoT intrusion detection
[18] Nimbalkar et al.	Effective reduction of feature count for accurate detection	Limited discussion on scalability for large datasets	Feature selection techniques for enhancing IDS in IoT networks
[25] Kumar et al.	Development of distributed intrusion detection system	Limited details on the specific methodology used	Distributed intrusion detection for DDoS attacks in IoT
[21] Zhou et al.	Innovative use of GNN for un-known attack detection	Increased time consumption for predictions	GNN-based intrusion detection using HAA and RWR techniques
[26] Shukla et al.	Examination of various machine learning methods in IoT	No explicit discussion of limitations	AI-based intrusion detection in IoT with focus on data selection
[20] Alsaedi et al.	Development of new dataset for IoT intrusion detection	Limited information on performance evaluation	Data-driven approaches for IoT and IIoT intrusion detection
[23] Kan et al.	Incorporation of PSO with CNN for improved reliability	Limited details on how cross-entropy loss is implemented	Adaptive PSO-CNN model for multi-type intrusion detection

The deep learning (DL) scheme described in paper [34] is something novel and effective with a three-level model. An architecture which consists of Convolutional Neural Network (CNN) along with a Bidirectional Gated Recurrent Unit (BiGRU) has been proposed here for the purpose of identifying organized accused (intruders). The evaluation mechanism of the BiGRU Poem is a feature all to itself. As a counterpart of this, the accuracy of this method can be further increased by utilizing a nature-inspired optimization method called the Wild Horse Optimization (WHO) as a meta-heuristic technique.

In [35] a token IDS-DL-DDoS Botnet detection method is unearthed via an innovative ML technology. The resulting solution exploits the mixture from the normal traffic data and the regular traffic data of a malicious type along with a scalable DNN designed specifically for robust detection of IoT botnet threats.

This extension is about intrusion detection systems (IDS) which is discussed in [36] when an ID super-parameters control system, termed HyConSys is presented. The network takes care of the extraction of significant data units of features that are readily labeled, and this involves the clustering process making use of the k-means algorithm. The system gains in integrity by involving the Proximal Policy Optimization (PPO) agent which supervises the IDS through adjustable learning and control functions.

In [37], an approach of DL-based anticipatory ANN is presented that achieves better accuracy k-barrier computation for identification and mitigation of intrusion attacks on machine learning models. This approach takes advantage of four latent features, which are two regions of interest and two different frequency bands that are utilized in active sonar operation with a Monte Carlo simulation for the neural network elements and hence improve their estimating capability.

In [38], a deep multilayer identification approach is introduced for intrusion detection, which operates in two stages: a pure understanding of what as well as a form of invasion it is. The quality of the final continued to look into introducing an oversampling approach is apply. Large numbers of tests are the main component of this approach to find out the efficiency across numerous scenarios and the application of oversampling techniques.

The authors of IDS DDoSNet in [39] have presented a novel IDS, which contains a PSO-based feature extraction algorithm and is therefore intelligent.

In [40], an animal detection system is suggested which is both low-cost, robust & scalable, based on DL and computer vision technology. The PIR system is powered by Raspberry Pi units, which crunch the image data to detect the presence of animal life. The MobileNetv2-SSD method is aimed at initial achievement of target detection and ResNet50 model mixed with Triplet Loss training is used to improve the accuracy of animal recognition.

Moreover, it is discussed in [41] that a developed DL-based methodology for automatic classification of animals is a better approach than the existing one. This system involves an alert unit which is attached to the deep CNN with an animal repulsion circuit. To extract the features from complicated images, deep CNNs are employed and afterward these features processed by means of Cross-CNN techniques. After the feature extraction is completed, the DL method utilizes the extracted features to precisely classify various animal types.

In the world of the Internet of Things (IoT) the ability of phishing has inspired a repertoire of devices that are centered on apprehension and countermeasures. Mughaid and colleagues [42] designed an ML algorithm that splits the training and testing data for the identification of phishing attacks under the electronic mails. They compared three datasets to reach high precision detection rates. Abdulrahman et al. [43] created an ML method based on Random Forest classifiers to distinguish phishing sites and verify its benefits using a validity test.

Jain and Gupta [44] developed the PHISH-SAFE structure with ML technology that involved URL features. The system examines fourteen URL attributes to establish the authenticity of web pages, by using Naive Bayes classifier and Support Vector Machine classifiers, on a big dataset of phishing and layered links. Huang et al. suggested a novel scheme for phishing website recognition through capsule based NN that consists of multi-layered processing to do the multi- dimensional feature extraction from the URLs.

A study of the author that was done in [45] concentrated on key characteristics for phishing detection, employing the Fuzzy Rough Set Theory to pick the best features from benchmark datasets. The features were subsequently experimented on the usual classification algorithms for phishing detection purposes. In his work, Jain and Gupta [46] recommended a visual similarity and link patterns comparison to monitor fraudulent phishing activities in e-banking and commercial websites.

Azeez and colleagues [47] suggested an automatic whitelist detection approach of phishing. The links scrutinized included actual and visual links that were compared against a predefined set of trusted sites to ascertain their legitimacy. Additionally, [48] by Conghui et al. proposed a Convolutional Neural Network-based email phishing detection system (CNNPD) which categorizes the emails into phishing emails and normal emails. An innovation which employs the MFO-RELM for detecting threats by cyber-attackers was showcased in article [49] as a viable solution, proving its usefulness in detecting numerous threats involved in IoT network. The research work done by Ruiz-Villafranca, and his team is an instance of the application of MECInOT - it is an example of an open-source framework which constructs testing environments for general purpose IoT platforms and, employing tree-base ML algorithms, detects intelligent attacks on the constructs.

One of the representatives of the TON Center, Rookard and Khojandi [50], proposed to apply reinforcement learning-based network ISC that is based on huge database which is TON-IoT and Deep Q-Network (DQN) model to allow IoT systems to identify cyber-attacks effectively. To conclude, Mengash et al. [51] applied a method level integrator of an SRO framework and an ML enabled cybersecurity technique that developed a system to identify instances of cyberbullying on social media platforms (SRO-MLCOSN).

III. Methodology

This paper presents and elaborates a sophisticated approach to the methodology of intrusion detection in the IoT device environment. The method starts from pre-processing Tun-IoT dataset to create a quality input (Fig. 1). The obtained data is then run through a feature selection method to separate informative attributes, which are then used to train a convolutional neural network. The algorithms integrated into CNN are applied to optimize performance. A few Grey Wolf Optimization have been used to enhance the model: first of all GWO which works to find the best values for parameters. The next two, Modified Grey Wolf Optimization has been designed explicitly in this study to select the best features and the best parameters for the previous select. Secondly, Advanced Grey Wolf Optimization has been implemented to optimize the Grey Wolf Optimization performance by using the advanced algorithm structure.

The Advanced Modified Grey Wolf Optimization combines the strengths and mitigates the weaknesses of both MGWO algorithm, i.e., new pattern search and existing proven computational pathway and is desined ot strie a delicate balance between these two, ensuring both robustness and adaptability of the model. Each variant has been tested against specific performance indicators to assess efficiency and accuracy. that helps to increase the accuracy and reliability substantially of the Intrusion Detection Systems in IoT. A model with advanced computational methods not only increases the accuracy of detection but is also well-interpretable and usable for the end users.

As a result, it is also anticipated to show a clear improvement in the detection and response time (real-time detection), false positives and negatives, and response time dealing with a variety of increased security threats such as zero-day attcks, enhancing the robustness of the system. These enhancements are essential in the case of deploying in security criticl conotexts, as they will ensure efficient protection against attacks.

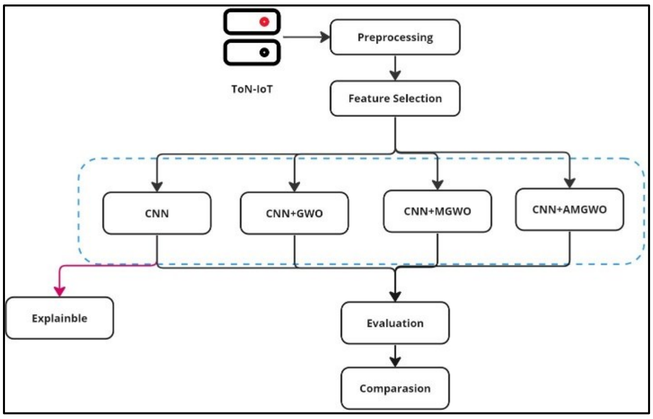

Fig. 1. Proposed Scheme

A. ToN_IoT Overview

The ToN_IoT datasets [52] introduced and referenced in and are cutting-edge and designed to facilitate the testing and evaluation of cybersecurity mechanisms in Internet of Things and Industrial Internet of Things systems. The naming ToN_IoT encapsulates the all-encompassing coverage of telemetry data from both IoT and IIoT sensors, including operating system data from well-supported platforms, such as Windows 7 and 10, Ubuntu 14 and 18 TLS, and also, extensive network traffic data, specifically making them an optimal resource for security analysis. These datasets result from a comprehensive setup from UNSW Canberra Cyber, School of Engineering and Information Technology located at the Australian Defence Force Academy. Through robust efforts, the data collection procedure is structured in parallel processing mechanisms controlled environment in real-scale networks established purposely to meet these requirements. An experimental device to emulate the IoT and Industry 4.0 settings was developed by the IoT Lab.

A variety of components are incorporated, such as virtual machines, physical set-systems, hack platforms, and cloud and fog computing technologies, and meticulously orchestrated to emulate the vast network architecture intricacies, as typically observed in IIoT environments. The heterogeneity of these records carries normal operational and cyber scenario data ranging from different forms of cybersecurity threats as well. Attacks triggered different hacking techniques deployed in different parts of the network such as web apps, IoT gateways, and computers. The hacking techniques employed are mainly DoS, DDoS, and the ransomware attacks designed specifically to determine the systems` resilience and adaptability incorporating every common and emerging cybersecurity threat. This comprehensive dataset does not only allow an exceptional platform to evaluate various AI implementations in cybersecurity but also is the forefront data in the development and validation of AI algorithms and other security solutions. Therefore, the dataset caters not only to the research domain but as well as the practitioners in conducting studies that ensure IIoT augmented cybersecurity malgines the increasing cybersecurity threats.

B. Preprocessing

The step of preprocessing is the most significant of all steps, here the data is being prepared in such a way that that it is suitable for the model training and the subsequent analysis. The process starts with dividing the dataset to come up with the feature vectors and the target. The feature constant ′X′ is derived by dropping the ’label’ and type columns from the imputed dataset; as a result, it isolates the set of variables associated with the intrusion detection system. Concurrently, two label sets are created: y1 is designed for the purpose of a classification task, whereas, y2 is created with multi-task learning scenarios or alternative analyses.

The subject y1 is screened to find out the number of a special classes, which determine an architecture of the model via an output layer’s dimensionality. The number of classes precisely is what determines the organization of the final classification layer of the neural network. This is also a precondition especially for those prediction methods such as one-hot-encoding which follows.

After separating features and labels the feature set X undergoes normalization which is borne out. A

MinMaxScaler is used typically to bring the ranges of features to fixed range, for example [0, 1]. Therefore, when scaling the gradient descent optimization algorithm that is used during a deep learning model's training, this will occur more efficiently by working its way on the smoother error surface.

Furthermore, through close checking of X, the existing presence of NaN (Not a Number) is checked, and if it is detected then this may lead to a reduction in the learning process. NaN values may be indicating incomplete data sets or errors in the data collecting processes. Therefore, most likely, they should either be imputed or eliminated. the model evaluation operation will be carried out by the evaluate_classifier, a custom function defined for this purpose. This function will be responsible for measuring the accuracy of the classifier on the test set using variety of metrics such as classifier accuracy, error rate, confusion matrix, and a detailed report on classification containing precision, recall, F1-scores for each class. This plotting routine also creates a confusion matrix to help interpret model performance with a matrix and also sensitivity and specificity for a more precise observation. Finally, we come to the data splitting into the training and testing subsets with the testing one being about 20% of the original dataset. This segregation or put this separation is paramount for the model assessment on the unseen data, thus producing a good speculation of the model's generalization capacity.

C. *Feature Selection*

Feature selection is a technique that helps to wade out the most significant features that can contribute to the performance of the model in prediction among the features which contain data information. This is called a very important step, as it not only lets the model achieve higher performance level by mean of cutting redundancy and noise but boosts computational efficiency.

With a hybrid objective function that combines F-score with features of statistical distribution such as kurtosis and skewness [53]-[56] and normalizes by the standard deviation, the memory of each feature will have its own weight in the model. The features that are measured to have higher F-scores are often considered to have better predictive power with respect to the dependent variable.

Although features with extreme values of kurtosis and skewness may aid in the stability or reduce the risk of bias in the model, they are subtracted from the F-score in the fitness calculation as they may also introduce bias into the model or even instability. For the sake of fairness, the variance normalization has been used to escape the role of one feature becoming too large and, thus, getting more prominence in selected features.

Next, forward selection works as a wrapping that underlies a hybrid objective function of the feature selection by iterative method. Through trial and improvement, vitality is added gradually; one feature added at a time until the innocuous model is maximized in terms of performance demonstration, or a pre-defined number of stable cycles are reached.

Among features, metrics including F-score, skewness, kurtosis, and variance are computed to provide attitude to multi-dimensional distribution changes of these metrics before feature selection and afterward.

The figures presented, identified as Fig. 2, Fig. 3, Fig. 4, and Fig. 5, violin plots to visualize the distribution of various statistical metrics across all features compared to those that have been selectively retained, referred to as "Leave Out" features. Violin plots are particularly effective for this purpose as they provide a clear illustration of the data's distribution and density across different values, represented along the y-axis. The width of each plot at different points along the y-axis reflects the concentration of data points, offering a visual representation of frequency.

In Fig. 2, which focuses on the distribution of F-scores, the violin plot reveals a notably narrower distribution for the selected features as opposed to the broader distribution observed across all features. This narrowing indicates that features with lower F-scores, which likely contribute less to predictive accuracy, have been effectively filtered out during the feature selection process. Such refinement highlights the effectiveness of the selection criteria in enhancing the model's focus on more predictive elements, thereby streamlining the feature set to improve model performance and reduce computational complexity.

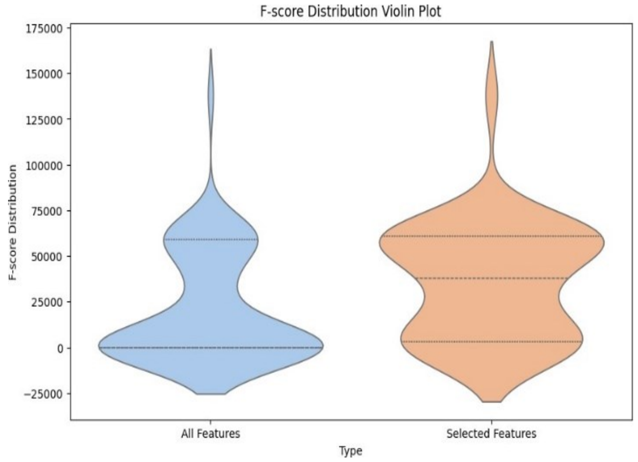

Fig. 2. Distribution of the F-scores

In Fig. 3 This diagram illustrates the skewness in the distribution of feature values. The skewness measure helps in understanding the asymmetry of the data distribution in relation to the normal distribution. The plot suggests that selected features tend to have a distribution that is more symmetric compared to the complete set, which may contribute to better model performance by focusing on more statistically balanced features.

In Fig. 4 the plot focuses on the kurtosis of feature distributions. Kurtosis is a measure of the "tailedness" of the probability distribution. The selected features show a sharper peak in their distribution, which might indicate a greater outlier presence or heavier tails, factors that can influence model sensitivity and robustness.

In the next Fig. 5 the following can be seen: the variance of the features, where a higher variance in the selected features compared to the full set suggests a preference for features with more variability in the data, potentially enhancing the model's ability to capture more complex patterns.

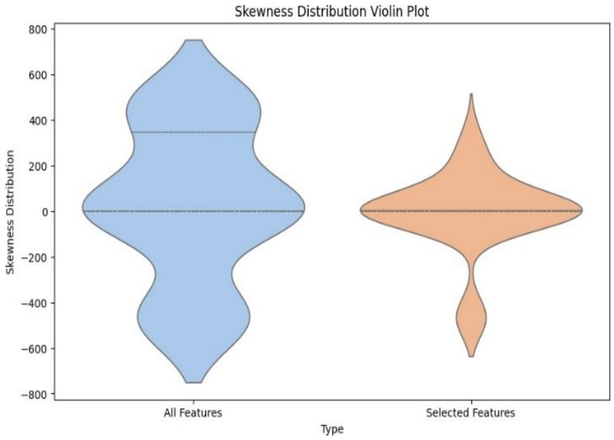

Fig. 3. The Skewness Distribution

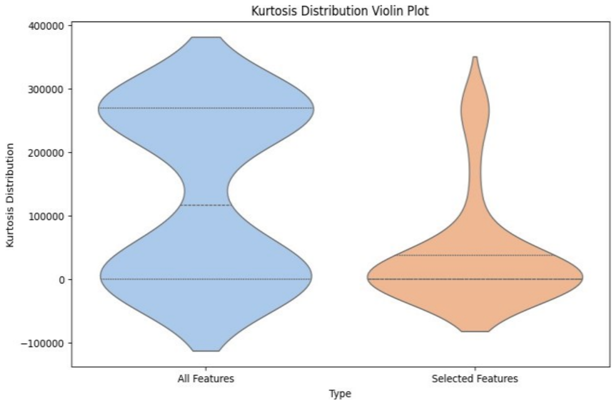

Fig. 4. The Kurtosis distribution

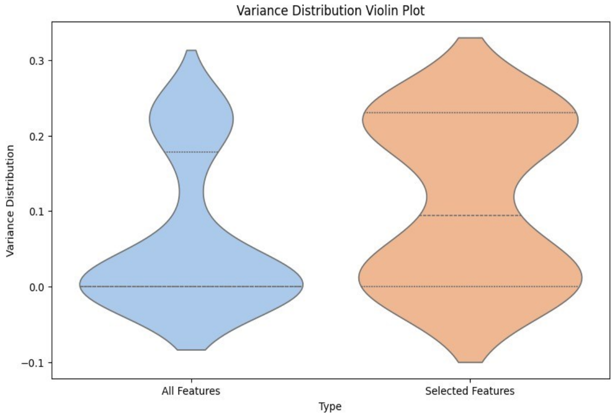

Fig. 5. Variance distribution

The apotheosis of this process comes after weaving the selected feature columns into this new data frame into which only the features which we required are included. These two lists now make up the training dataset. The labels for multi-assembly status are concatenated and are saved to a CSV file for future use. Thus, the feature selection phase is completed, and our predictive models will be reliable and robust.

D. CNN Architecture

The CNN architecture, described in [57], is prepared to thoroughly analyze and process the extracted relevant characteristics highlighted above. This architecture is laid out according to a series of layers, all of which are programmed to perform several individual tasks that together allowed the model to look for intricate patterns in the dataset. The initial layer of the CNN is the flattening layer, containing predetermined filters and kernel sizes. Convolutional layers operate by gliding the kernel across input data and, in doing so, scan these data points for local features or patterns. The filter and kernel widths are critical architecture parameters as they determine how broad or narrow the composite feature field will be, allowing the model to focus on the pertinent pieces of data. The activation layer is next using a linear rectified unit function, usually. This layer introduces non-linearity into the input, making it simpler for the model to detect complicated and non-linear relation characteristics. This will allow CNN to generalize better and then learn from a varied input dataset. The max-pooling layer is the next. This layer decreases the dimensions of the obtained maps previously generated by a variety of other layers. This is accomplished by keeping the most pertinent characteristics detected by the preceding layer against the variant input. This approach becomes more efficient in subsequent steps because fewer parameters are required to compute the subsequent layers. A flattening approach is implemented. Since the preceding step provides spatially arranged characteristics, they must all be refocused into an input vector to configure it.

Next, the fully connected layer is added after the previous layer. Also called the dense layer, this serves as the ‘control center’ where the weight of how important the feature is computed in making predictions. Reduction in the number of neurons is done in the layer-wise manner, which turns into a model, and reduction in the number of high level features leads to a very simple model called an output layer. The last layer contains the following layer in this CNN, which is terminally ending layer containing softmax activation function [58]. This function develops the requirement of the input that needs to do the classification of the prediction. Further, it develops a probability distribution over the classes. Probability serves or shows the confidence of the input form to the given data how the layer will make the classification of the feature within the layer. This essential in multi classification class where the input features may be classified in more than one class. Finally, compilation of the model to choose the gradient function and optimizer which acts as the guide during compilation. In deciding the loss the guide predictor that measures how the model ought to maintain during the detection level the loss is used as a guideline to the optimizer which helps the loss addition to the model to minimize the loss. In conclusion, the described method gives a step to step description of CNN architectural impact in terms of classification of inputs of data complex. The use of the linear and a non-linear layer processed in an effective way due to the high dimensionality of the data which can easily identify patterns with high accuracy.

E. CNN+GWO Architecture and Parameters

In reference [59], the authors propose integrating Grey Wolf Optimization to enhance the architecture of Convolutional Neural Network. GWO is integrated to effectively finetune critical hyperparameters of the network based on its robust capability for global search and optimization. It mimics a social hierarchy and hunting strategy of the grey wolves in the wild. Such critical optimization can explore complex hyperparameters space of CNNs to optimize and achieve high performance without significant computational costs. GWO has multiple roles based on the social structure of wolves, and proposed solutions can take the form of an alpha, beta, or delta. These roles direct the optimization procedure, with the alpha representing the best current solution. In each subsequent iteration of GWO, the positions of candidate solutions are updated, and their movement represents wolves' movement towards the prey. In this metaphorical scenario, GWO helps find optimal configuration and also ensures a search that avoids local minima. The candidates are different sets of hyperparameters, including the number of filters and sizes of filters in the convolutional layers of CNN. The optimal configuration of hyperparameters will result from dynamic adjustments made to the positions of each candidate. The adjustment is done through an iterative process of GWO per their fitness based on their performance in predicting outcomes using CNN. The process continues until an optimal solution is met, which is the best balance of filter numbers and sizes.

The range of values explored during this optimization process, particularly in terms of the diversity of filter sizes and the depth of convolutional layers, is essential: in order for the CNN to be able to learn to differentiate between features in the input data regardless of whether they are large-scale structures or fine details, the optimization process must establish a method to ensure that the desired diversity of filter dimensions is achieved. In other words, one of the main goals of employing GWO during optimization is not to end up with the best CNN architecture proven to be the most accurate, but to develop one that can also generalize better across datasets and problems. Once the optimization process is complete, the best-performing configuration is selected using the results of GWO. Generally, this configuration features widely optimized numbers of filters, typically ranging from 64 to 256, and kernel sizes spanning 2×2 pixel regions up to 5×5. Once the ideal configuration is determined, the model is trained on the entire dataset for validation and testing. The finalized, GWO-optimized model is then tested extensively on novel data to ensure that it performs similarly on training data and generalizes well. By following this thorough procedure and utilizing the results of the bio-inspired GWO optimization method, the CNN model created for this application is able to achieve high generalization capabilities and robust performance, which is necessary for most pattern-detection applications one might deploy it in. Ultimately, the incorporation of the GWO into the training process uses the hyperparameter space to better refine the architecture and function of the model for the desired application. Therefore, the natural metaheuristic optimization method discussed herein can be used to increase the interoperability, efficiency, and sophistication of deep learning by creating better optimized models.

F. *CNN+Modified GWO Architecture and Parameters*

In many other future studies of searching for better performance in CNNs, MGWO [60] can be utilized as a replacement to optimize those network hyperparameters, such as the number of convolutional filters and the size of the convolutional kernel. The method used in the study is intended to modify the original algorithm of GWO in the domain of deep learning by incorporating new fixtures to the initially and update mechanisms. Therefore, the method is anticipated to be efficient and accurate if it is implemented. Convolutional one of CNN starts with the MGWO deciding the ideal number filters and the kernel size. This layer is the central that allows for extraction of the features from the data inputs. After the result, the introduction of ReLU nonlinearity is the next. Max pooling layer is presented herein to discard the non-critical features and it makes focus on the main torments. After the loss function and the non-linear activation function, the flatten output goes to a sequence of dense layers with the last one being a softmax layer, which predicts the probability distribution is over two classes.

Since MGWO is inspired by a social structure of grey wolves' group and the hunting setting, which gives the best solution for the given interval, targeting on the policyholder problems and deciding the payout and the number of jobs, also filters 16-256, the size of kernel 2-5 is set. These confines are used for both inclusion and elimination, thereby enabling search space that can cover all reasonable values and at the same time be focused on the most reasonable among them. However, simultaneously, each of the wolves and equivalent solution candidates have been assigned smart start scripts. These scripts offer conservative addition of kernel size and number of filters and estimate wolf's position in the next iterations based on their fitness, which is mobilized as an attribute of the CNN when the same hyperparameter is trained. The modifications performed by wolves are made through vertical dynamic adjustment coefficients which modified by alpha, the top-notch solution, beta and delta which are female for the rest. The cycle of adaptation is the most vital part of the evolutionary capacity of the GWO so that there is a balance of population based on the search space search and most promising solution. The cycle is repeated by the network so that the wolves are updated and this iterative process leads the wolves to the best configurations that are more robust for the CNN. At the end of the cycle, the best solution is selected and the best array of filter sizes, number of filters, and a kernel size is determined which make sure that all through the training and validation of CNN it can achieve the model accuracy.

For the CNN+MGWO, having already determined the best parameters for the optimal environment, that CNN is trained over some a set number of epochs and with a specific batch size, and its performance is likewise monitored. Using training loss, validation loss and accuracy curve gives the network learning and visualize the training process each and every epoch. These illustrative instruments are able to be trustworthy on the matters is the model's convergence and also effectiveness of the training level. To sum up the CNN+MGWO, therefore, it puts forth that CNNs' potent characteristic in the field of the feature learning side and, in the other side, MGWO has a very strong optimal technique. Thus an embracement of hybrid techniques not only render deep learning networks architecture-wise, in a highly adaptive and efficient way but also provides a good reference for the smart utilization of the bio-inspired algorithm in the field of artificial intelligence.

G. CNN+Modified GWO + Advanced GWO (AMGWO) Architecture and Parameters

The architecture of the Convolutional Neural Network (CNN) is significantly enhanced through the integration of two sophisticated optimization strategies: the Modified Grey Wolf Optimizer (MGWO) and the Advanced Grey Wolf Optimizer (AGWO) [61]. This dual optimization strategy not only aims at refining the primary architectural elements of the CNN, such as the number of convolutional filters and the kernel sizes, but also meticulously fine-tunes these elements to achieve outstanding accuracy and precision during training phases.

In the initial stages of the CNN development, the architecture employs a convolutional layer whose configuration is dynamically shaped based on the outcomes derived from the AGWO method. This layer forms the foundation of the network by extracting crucial features from the input data. The next element is the non-linear activation function, which is typically Rectified Linear Unit responsible for enabling the network to learn complex patterns from the data. The third layer is the max pooling layer which reduces the spatial dimensions of the data. This not only aids in reducing computational complexities but also ensures that the most essential features are preserved. This enhances the model's generalization power by utilizing the model's strengths effectively. The process continues as the vectors and tensors get flattened and converted into a 1-D vector. This is essential in preparing the data for the final classification. The data then proceeds through several fully connected layers that reduce the booming dimension of the data until it reaches the softmax layer. This layer is important since it outputs a probability distribution over the various classes, enabling quantifiable inferences into the network's predictions. The AMGWO is an application of intelligent adaptive mechanisms that allow on-the-go adjustments based on the CNN's performance landscape, is an advanced innovation derived from the normal GWO's modification and addition.

Inspired by the social structure of grey wolves and their hunting dynamics, the AMGWO utilizes a pack of candidate solutions to effectively navigate the hyperparameter space. Each candidate is referred to as a 'wolf' and is tasked with adjusting their position in the search space through a set of heuristic rules that rely on their different performances or 'fitness'. This process allows the network to dynamically identify optimal parameter spaces hence balancing the exploratory landscape and that of exploitation, continually optimizing the performance.

The AMGWO algorithm executes in a pre-defined hyperparameters range which includes the filter number's adjustment within the 32 to 256 range and kernel size in terms of pixels of 2 times 2 to 5 times 5. The optimization is conducted in an iterative manner where the position of wolves is repositioned after considering their updated fitness score at each step. The aim of this process is to converge on a set of hyperparameters that results in an optimal classification of the network concerning the validation datasets. Following the identification of an optimal setting, an elaborate training is developed, which is performed over many epochs.

Under this training phase, the CNN performance is monitored continuously using the training and validation loss and accuracy visualization. This analysis is vital in understanding how learning is happening in the network and hence to make an adequate adjustment to facilitate proper convergence without overfitting.

Finally, the CNN+AMGWO is an elaborate evaluation based on a comprehensive metric-based evaluation. The evaluation is executed using a diverse performance metrics that evaluate the ability of the model using the loss and accuracy metrics, and the performance of the classifier. This is an essential evaluation metric as it is vital in determining the practicality of the model in real-world scenarios. It ensures the network is reliable and capable of achieving desirable solutions over various datasets and conditions. Thus shows the effectiveness of the CNN model reinforced by AGWO is effective in complex pattern recognition.

IV. EXPERIMENT RESULTS

The results obtained from applying different optimization techniques sequentially—Standard Grey Wolf Optimizer, Modified GWO, and Advanced Modified GWO – reveal optimization of the performance metrics of the CNN model. The metrics as shown in a table below reveal how each optimization technique (SGWO, MGWO, and AMGWO) improves the accuracy, error rate, sensitivity, and specificity of the model.

The accuracy of the model is an indication of how many of the model's predictions are correct. The trend shows a progressive increase of accuracy from 97.4% to 97.4%, 97.5% & 97.9% as the optimization techniques are applied. The increase in accuracy from 97.4% for the standalone CNN model to 97.9 compounded with AMGWO demonstrates how GWO methods can be used to fine-tune a model for various datasets.

The error rate, which is the inverse of accuracy, shows how often the model is making mistakes in predictions. The error rate reduced from 2.59% to 2.0%, 2.04%, and 2.02%, revealing that the model is making fewer mistakes and has been refined.

The sensitivity is the true positive rate showing how well the model can spot positive instances. Sensitivity for the CNN model is 97.3% increasing to 98.5% as AMGWO is applied, which demonstrates that the model predicts actual positives without missing them.

The specificity is the true negative rate which indicates the model's ability to spot negative instances. It increased from 97.5% through. It shows that the model misidentifies few negatives as positives or classes well the class labels leading to less false alarms. All the above improvements are not just numerical improvements but rather developments that make a model strong. Such improvements are critical for models that are to be utilized in fields where there should be high reliability. This includes fields such as medical diagnostics, autonomous or self-driving vehicles, and cybersecurity in such a case as shown in this article. These methods of algorithms are viable and applicable not only in ensuring the highest levels of accuracy are attained but also ensuring that the models can distinguish a true attack from activities that are not in a complicated pattern.

TABLE III. SUMMARY OF EXPERIMENTAL RESULTS

Model	Accuracy	Error Rate	Sensitivity	Specificity
CNN	0.974	0.0259	0.973/0.975	0.975/0.973
CNN+GWO	0.977	0.0228	0.964/0.993	0.993/0.964
CNN+MGWO	0.978	0.0211	0.975/0.983	0.983/0.975
CNN+AMGWO	0.979	0.0202	0.976/0.985	0.985/0.976

Fig. 6 presents a bar graph used as a local explanation tool within the context of Explainable AI to clarify how a convolutional neural network (CNN) model predicts a 'class attack'. This type of visualization elucidates the influence of various input features on the model's decision-making process. Each bar indicates the weight or importance of a specific characteristic of the input data in determining whether the input is classified as an attack.

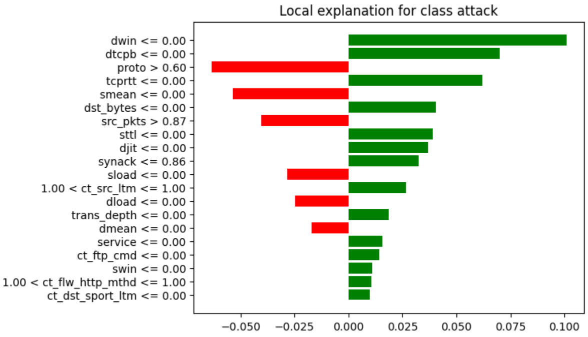

Fig. 6. Explainable AI

The green bars in the graph depict features that contribute positively towards the classification of an input as an attack. These include features such as ‘synack <= 0.86’ and ‘src_pkts > 0.87’, which are represented by large features with positive weight, meaning that once the actual input feature values exceed this threshold, it drastically increases the probability of predicting an attack. These bars’ size further explains how influential the feature is on the model’s final output.

Additionally, red bars demonstrate that the feature has negative influence on the model’s accuracy in its prediction of the input, thus referred to as a scenario that mimics an attack or one of least concern. Red features, such as ‘ct_state_ttl <= 1.00’, provide a substantial size feature with a negative effect, which reduces the probability of predicting an input as an attack whenever the feature has a value. The feature thresholds, for instance, ‘<= 0.00’ and ‘> 0.60’ next to each feature, shows the point at which the influence of the feature changes from positive to negative and vice versa.

Additionally, for the categorical feature, expressions such as ‘service = 0’ show how a particular category affects the prediction of an attack. This figure is important in the experimental setup since it shows the bridge between model outputs and the humanly understandable reason. It demonstrates the most influential features in the model’s decision virgin. This reason helps in validating the model’s prediction and might reveal possible areas of improvement in the model.

By analyzing how different features affect the model's predictions, researchers can identify which aspects of the data are most predictive of attacks, potentially leading to improvements in feature engineering and model training. This level of interpretability is crucial in applications like cybersecurity, where the justification for a model's decision to flag an activity as an attack is as important as the accuracy of the prediction itself.

The discussion section of our research paper further deeply explores the implications of our findings. Similarly, we revisit the core research questions and provide an overview of how our findings are positioned in light of existing knowledge in the artificial intelligence and cybersecurity domain.

Indeed, by considering the various application of GWO strategies to enhance the accuracy of CNN models and optimize their operational efficiency, our findings offer critical insides into how this can be achieved under real-world scenarios. Our empirical analysis portrays an improved accuracy performance with CNN models following the application of GWO variants – Standard GWO, MGWO, and AMGWO. Already a strong standalone model with accuracy performance of 97.4%, the successive application of GWO strategies systematically enhance the model performance

Notably, it’s the overall improvement in the generalization of these models that underlines the significance of models’ performance in diverse and dynamic security threats in ensuring the balance of sensitivity and specificity. Notably, the achieved balance provides a comprehensive approach to the elimination of bias, naturally present in most datasets, which can skew model output and cause less reliable systems.

Additionally, the reported improved sensitivity in MGWO and AMGWO used models prove to be a critical attribute. Particularly, improved sensitivity means the models can identify and classify true positive instances effectively. This is a critical attribute in cybersecurity, given oversight or wrongly classifying a true threat without adequate sensitivity rates can be extremely costly. Significantly, it offers a judicious and responsive security approach, emphasizing that the optimized models are capable of minimizing false positives.

The scope of these technical advancements is, from a practical perspective, immense in the various cybersecurity uses. For instance, in intrusion detection systems , this level of accuracy and rate of errors indicates that the system can detect minor subtle and sophisticated cyber threats with improved accuracy. Such capability can be of significant importance in a scenario where the agents of an attack on a machine apply sophisticated methodologies, evading unsophisticated mechanisms.

The model’s increased sensitivity level enables the accurate detection of vague anomalies rays that may signal towards a security breach and enhanced specificity ensures that the daily network operations are not falsely detected as threats, reducing the rate of false alarms that may cause disruptions. The applications of this CNN model optimization are not limited to the traditional IT setup but go further to include the security and monitoring of IoT devices and the financial systems.

In the latter case, the model optimization will be vital in the detection and fight against fraud and theft which will be a significant improvement in the financial transaction domain. Generally, it is possible to assert that the CNN model optimization facilitated by evolutionary algorithms like GWO, not only improve the accuracy levels and efficiency but dramatically widen the application areas and significance.

Therefore, this research highlights the importance of continuous improvement on the machine learning algorithms considering the growing levels of cyber insecurity. It perfectly captures the importance of the technical improvements triggered by the machine learning innovations and the cybersecurity measures to the many interconnected data systems globally.

V. Comparative Analysis

The table presents a detailed comparative analysis of the proposed Adaptive Particle Swarm Optimization Convolutional Neural Network (APSO-CNN) algorithm with several related works in the field of IoT network intrusion detection. All key performance metrics were considered for comparison: accuracy, precision, recall, F-score, training time, and training time.

APSO-CNN had the highest accuracy among all models considered, which implies its ability to identify normal and malicious activities accurately in the IoT network. Thus the algorithm has a robust performance in all kinds of intrusion scenarios. Further, APSO-CNN achieved high precision and recall. High precision indicates that the algorithm had a low false positive rate and rarely marks the normal network activity as intrusion.

Similarly, High recall indicated APSO-CNN had high power to detect most of the intrusion activity with little false negative. In addition to a balanced F-score, APSO-CNN implies that there is no possibility of one standard surpassing the other.

In terms of computational time efficiency APSO was more effective compared to other methods. Even though the Network time epitomizes the highest training time, it is not too slow for the high accuracy it demonstrates. This is an efficient testing tree; therefore, an optimal method can be used for testing real-time applications.

The overall performance and reliability comparison in a 3-type IoT network intrusion task portrays a high and efficient reliance on the proposed method. It depicts the high-performance evaluation surpasses all the compared methods with robust performance. The paper confirms the high differences the proposed method has against the current methods. Assessment Measuring System.

VI. Conclusion

The current study integrates deep learning and evolutionary computation to present the performance difference of CNNs depending on the Wolves Optimizer variants. The sequential implementation of Genetic Algorithm, Modified GA, and Advanced and Modified GWO incorporated in the CNN were examined to visualize how each optimization algorithm dynamically adjusts model parameters, leading to incremental enhancement in predicting accuracy and deterioration in error rate. As denoted here, these optimization strategies are all intertwined, demonstrating their contributions, which lead to the regular staircase in terms of maximizing accuracy.

The AMGWO-optimized CNN gained the best achievement, the smallest error rate and highest accuracy and, therefore, supports the suggested hypothesis that nature-inspired algorithms exhibit capacity advances in the course of learning machines. The implications of all these outcomes are vital. The optimized model specifically showed equally balanced sensitivity and specificity, marking every model as highly reliable and therefore applicable in the real world, such as health and safety professional scenarios, which require the best prototype that could be trusted. The optimized model has high sensitivity; therefore, it will help every case targeted and positively identified as required. The model too is specific, meaning identified cases are always on the positive side.

Overall, the integration of evolutionary algorithms with deep learning has a great promise in various areas. The utilization of GWO-based optimization for enhancing CNN models opens broad prospects for industries such as healthcare, finance, and cybersecurity. Specifically, more accurate models can ensure that patients receive proper diagnosis and treatment by detecting diseases with the highest reliability. In finance, CNN models improved by GWO-inspired algorithms can enhance the prediction of risks and benefits, support better decision-making, and identify frauds. In the field of cybersecurity, fine-tuning CNNs using GWO techniques will allow building more sensitive and specific intrusion detection systems, thereby ensuring proper protection against attacks.

Furthermore, the current paper provides valuable information on the nature and functioning of CNN optimization using GWO-inspired approaches. This information is essential for practical implementation and has great implications for the development of powerful and human-like AI systems. Further studies can explore the application of GWO-based optimization to other DL models outside CNNs, such as RNNs or Transformers, or discrete GWO techniques to address existing drawbacks of CNN models, such as a small number of samples or reduced computational efficiency.

Thus, research in this area can significantly contribute to the development of intelligent systems and the provision of more effective solutions for complicated problems in varied fields. The consistent use of language and the evident professionalism of the approach to presenting the topic underscores the clear reflection of research findings and their implications. In this way, the paper guides further research on the subject and demonstrates the potential of merging evolutionary algorithms with DL for the creation of strong and practical AI systems.

TABLE IV. TABLE OF COMPARATIVE ANALYSAIS

Reference	Key Contributions	Approach	Accuracy
Our work	Optimization of CNN using GWO variants	Evolutionary algorithms for optimizing CNN in cybersecurity	CNN : 0.974 CNN+GWO : 0.977 CNN+MGWO :0.978 CNN+AMGWO : 0.979
[19] Hindy et al.	Focus on MQTT-IoT-IDS 2020 dataset and bi-directional features	Case study on machine learning techniques for intrusion detection	RBFN showed superior performance with up to 89.7%.
[24] Abdalgawad et al.	Effective use of Bi-GAN and generative deep learning	Bi-GAN model for detecting cyberattacks in IoT networks	C&C-PartOfAHorizontalPortScan and C&C-HeartBeat, recorded the lowest F1-scores (0.93).
[22] Wahab et al.	Detection of intrusion data streams with drift detection	Online deep learning approach with data drift detection	-
[17] Islam et al.	Focus on quick implementation and effective handling of unknown events	Use of machine learning and deep learning algorithms for IoT threat detection	SVM showed the highest accuracy across NSL-KDD, IoTDevNet, and DS2OS datasets with scores up to 99.84%.
[16] Da Costa et al.	Comprehensive review of IoT intrusion detection techniques	Review of machine learning techniques for IoT intrusion detection	-
[18] Nimbalkar et al.	Effective reduction of feature count for accurate detection	Feature selection techniques for enhancing IDS in IoT networks	- Demonstrated that reducing the number of features through sophisticated selection methods could maintain or even enhance the accuracy and efficiency of IDS.
[25] Kumar et al.	Development of distributed intrusion detection system	Distributed intrusion detection for DDoS attacks in IoT	Experiments were conducted on a Tyrone PC with specific hardware specifications.
[21] Zhou et al.	Innovative use of GNN for unknown attack detection	GNN-based intrusion detection using HAA and RWR techniques	The proposed method demonstrated a reduction in classification precision by more than 30% in state-of-the-art GNN models, GCN and JK-Net, using UNSW-SOSR2019 dataset.
[26] Shukla et al.	Examination of various machine learning methods in IoT	AI-based intrusion detection in IoT with focus on data selection	-
[20] Alsaedi et al.	Development of new dataset for IoT intrusion detection	Data-driven approaches for IoT and IIoT intrusion detection	- binary classification CART achieved the highest scores in accuracy (0.88), precision (0.90), recall (0.88), and F-score (0.88). - multi-class classification CART again performed best with scores of 0.77 in accuracy, precision, recall, and F-score.
[23] Kan et al.	Incorporation of PSO with CNN for improved reliability	Adaptive PSO-CNN model for multi-type intrusion detection	The results, validated through five traditional indicators and accuracy statistics from 10 independent experiments, demonstrate that the APSO-CNN algorithm is effective and reliable for detecting multi-type IoT network intrusion attacks.

REFERENCES

[1] J. Jang-Jaccard and S. Nepal, “A survey of emerging threats in cybersecurity,” *Journal of Computer and System Sciences*, vol. 80, pp. 973–993, 2014, doi: 10.1016/j.jcss.2014.02.005.

[2] J. Drew, T. Moore, and M. Hahsler, “Polymorphic malware detection using sequence classi- fication methods,” in *IEEE Security and Privacy Workshops (SPW)*, pp. 81–87, 2016, doi: 10.1109/SPW.2016.30.

[3] G. Canfora, F. Mercaldo, C.A. Visaggio, and P. Di Notte, “Metamorphic malware detection using code metrics,” *Information Security Journal: A Global Perspective*, vol. 23, pp. 57–67, 2014, doi: 10.1080/19393555.2014.931487.

[4] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, “Malware images: visualization and automatic classification,” in *Proceedings of the 8th international symposium on visualization for cyber security*, pp. 1-7, 2011.

[5] H. Kang, J.-W. Jang, A. Mohaisen, and H. K. Kim, “Detecting and classifying android malware using static analysis along with creator information,” *International Journal of Distributed Sensor Networks*, vol. 11, p. 479174, 2015, doi: 10.1155/2015/479174.

[6] W. Han, J. Xue, Y. Wang, L. Huang, Z. Kong, and L. Mao, “Maldae: Detecting and explaining malware based on correlation and fusion of static and dynamic characteristics,” *Computers & Security*, vol. 83, pp. 208–233, 2019.

[7] W. Zhong and F. Gu, “A multi-level deep learning system for malware detection,” *Expert Systems with Applications*, vol. 133, pp. 151–162, 2019, doi: 10.1016/j.cose.2019.02.007.

[8] R. H. Jhaveri, A. Revathi, K. Ramana, R. Raut, and R. K. Dhanaraj, “A review on machine learning strategies for real-world engineering applications,” *Mobile Information Systems*, vol. 2022, no. 1, p. 1833507, 2022.

[9] D. Gibert, C. Mateu, and J. Planes, “The rise of machine learning for detection and classification of malware: Research developments, trends and challenges,” *Journal of Network and Computer Applications*, vol. 153, p. 102526, 2020.

[10] U. E. H. Tayyab, F. B. Khan, M. H. Durad, A. Khan, and Y. S. Lee, “A survey of the recent trends in deep learning based malware detection,” *Journal of Cybersecurity and Privacy*, vol. 2, no. 4, pp. 800-829, 2022.

[11] R. Ali, A. Ali, F. Iqbal, M. Hussain, and F. Ullah, “Deep learning methods for malware and intrusion detection: A systematic literature review. *Security and Communication Networks*, vol. 2022, no. 1, p. 2959222, 2022.

[12] S. K. Smmarwar, G. P. Gupta, and S. Kumar, “Android Malware Detection and Identification Frameworks by Leveraging the Machine and Deep Learning Techniques: A Comprehensive Review,” *Telematics and Informatics Reports*, p. 100130, 2024.

[13] A. Bensaoud, J. Kalita, and M. Bensaoud, “A survey of malware detection using deep learning,” *Machine Learning With Applications*, vol. 16, p. 100546, 2024.

[14] A. F. Agarap, “Towards building an intelligent anti-malware system: a deep learning approach using support vector machine (SVM) for malware classification,” *arXiv preprint arXiv:1801.00318*, 2017.

[15] J. Zhang, Z. Qin, H. Yin, L. Ou, and K. Zhang, “A feature-hybrid malware variants detection using cnn based opcode embedding and bpnn based api embedding,” *Computers & Security*, vol. 84, pp. 376–392, 2019, doi: 10.1016/j.cose.2019.04.005.

[16] K. A. Da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, “Internet of things: A survey on machine learning-based intrusion detection approaches,” *Computer Networks*, vol. 151, pp. 147–157, 2019, doi: 10.1016/j.comnet.2019.01.023.

[17] N. Islam *et al.*, “Towards machine learning based intrusion detection in iot networks,” *Computers, Materials & Continua*, vol. 69, pp. 1801–1821, 2021, doi: 10.32604/cmc.2021.018466.

[18] P. Nimbalkar and D. Kshirsagar, “Feature selection for intrusion detection system in internet-of- things (iot),” *ICT Express*, vol. 7, pp. 177–181, 2021, doi: 10.1016/j.icte.2021.04.012.

[19] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, and X. Bellekens, “Machine learning based iot intrusion detection system: An mqtt case study (mqtt-iot-ids2020 dataset),” in *International Networking Conference*, pp. 73–84, 2020, doi: 10.1007/978-3-030-64758-2_6.

[20] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems," in *IEEE Access*, vol. 8, pp. 165130-165150, 2020, doi: 10.1109/ACCESS.2020.3022862.

[21] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, I. Kevin, and K. Wang, “Hierarchical adversarial attacks against graph-neural-network-based iot network intrusion detection system,” *IEEE Internet of Things Journal*, vol. 9, pp. 9310–9319, 2021, doi: 10.1109/JIOT.2021.3130434.

[22] O. A. Wahab, “Intrusion detection in the iot under data and concept drifts: Online deep learning approach,” *IEEE Internet of Things Journal*, vol. 9, pp. 19706–19716, 2022, doi: 10.1109/JIOT.2022.3167005.

[23] X. Kan *et al.*, “A novel iot network intrusion detection approach based on adaptive particle swarm optimization convolutional neural network,” *Information Sciences*, vol. 568, pp. 147–162, 2021, doi: 10.1016/j.ins.2021.03.060.

[24] N. Abdalgawad, A. Sajun, Y. Kaddoura, I. A. Zualkernan, and F. Aloul, “Generative deep learning to detect cyberattacks for the iot-23 dataset,” *IEEE Access*, vol. 10, pp. 6430–6441, 2021, doi: 10.1109/ACCESS.2021.3140015.

[25] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, S. Garg, and M. M. Hassan, “A distributed intrusion detection system to detect ddos attacks in blockchain-enabled iot network,” *Journal of Parallel and Distributed Computing*, vol. 164, pp. 55–68, 2022, doi: 10.1016/j.jpdc.2022.01.030.

[26] A. Shukla, S. Ahamad, G. N. Rao, A. J. Al-Asadi, A. Gupta, and M. Kumbhkar, “Artificial intelligence assisted iot data intrusion detection,” in *2021 4th International Conference on Computing and Communications Technologies (ICCCT)*, pp. 330–335, 2021, doi: 10.1109/ICCCT53315.2021.9711795.

[27] O. D. Okey *et al.*, “Boostedenml: Efficient technique for detecting cyberattacks in iot systems using boosted ensemble machine learning,” *Sensors*, vol. 22, p. 7409, 2022, doi: 10.3390/s22197409.

[28] M. M. Alani, “An explainable efficient flow-based industrial iot intrusion detection system,” *Computers & Electrical Engineering*, vol. 108, p. 108732, 2023, doi: 10.1016/j.compeleceng.2023.108732.

[29] S. Shruti, S. Rani, and G. Srivastava, “Secure hierarchical fog computing-based architecture for industry 5.0 using an attribute-based encryption scheme,” *Expert Systems with Applications*, vol. 235, p. 121180, 2023, doi: 10.1016/j.eswa.2023.121180.

[30] H. Albasheer *et al.*, “Cyber-attack prediction based on network intrusion detection systems for alert correlation techniques: A survey,” *Sensors*, vol. 22, p. 1494, 2022, doi: 10.3390/s22041494.

[31] A. Massaro, M. Gargaro, G. Dipierro, A. M. Galiano, and S. Buonopane, “Prototype cross platform oriented on cybersecurity, virtual connectivity, big data and artificial intelligence control,” *IEEE Access*, vol. 8, pp. 197939–197954, 2020, doi: 10.1109/ACCESS.2020.3034399.

[32] H.-C. Lin, P. Wang, K.-M. Chao, W.-H. Lin, and J.-H. Chen, “Using deep learning networks to identify cyber attacks on intrusion detection for in-vehicle networks,” *Electronics*, vol. 11, p. 2180, 2022, doi: 10.1109/ACCESS.2020.3034399.

[33] I. H. Sarker, Y. B. Abushark, F. Alsolami, and A. I. Khan, “Intrudtree: A machine learning based cyber security intrusion detection model,” *Symmetry*, vol. 12, p. 754, 2020, doi: 10.3390/sym12050754.

[34] A. F. Almutairi and A. A. Alshargabi, “Using deep learning technique to protect internet network from intrusion in iot environment,” in *Proc. 2nd Int. Conf. Emerging Smart Technol. Appl. (eSmarTA)*, pp. 1–6, 2022, doi: 10.1109/eSmarTA56775.2022.9935467.

[35] J. P. J. Shareena, A. Ramdas, and A. P. Haripriya, “Intrusion detection system for iot botnet attacks using deep learning,” *Social Netw. Comput. Sci.*, vol. 2, no. 3, p. 205, 2021, doi: 10.1007/s42979-021-00516-9.

[36] H. Han, H. Kim, and Y. Kim, “Correlation between deep neural network hidden layer and intrusion detection performance in iot intrusion detection system,” *Symmetry*, vol. 14, no. 10, p. 2077, 2022, doi: 10.3390/sym14102077.

[37] S. Muruganandam, R. Joshi, P. Suresh, N. Balakrishna, K. H. Kishore, and S. V. Manikanthan, “A deep learning based feed forward artificial neural network to predict the k-barriers for intrusion detection using a wireless sensor network,” *Meas. Sensors*, vol. 25, p. 100613, 2023, doi: 10.1016/j.measen.2022.100613.

[38] R. Qaddoura, A. M. Al-Zoubi, H. Faris, and I. Almomani, “A multi layer classification approach for intrusion detection in iot networks based on deep learning,” *Sensors*, vol. 21, no. 9, p. 2987, 2021, doi: 10.3390/s21092987.

[39] D. Srilatha and N. Thillaiarasu, “Ddosnet: A deep learning model for detecting network attacks in cloud computing,” in *Proc. 4th Int. Conf. Inventive Res. Comput. Appl. (ICIRCA)*, pp. 576–581, 2022, doi: 10.1109/ICIRCA54612.2022.9985524.

[40] P. C. Ravoor, T. S. B. Sudarshan, and K. Rangarajan, “Digital borders: Design of an animal intrusion detection system based on deep learning,” in *Proc. Int. Conf. Comput. Vis. Image Process.*, pp. 186–200, 2020, doi: 10.1007/978-981-16-1103-2_17.

[41] A Mughaid, S AlZu’bi, A Hnaif, S Taamneh, A Alnajjar, and EA Elsoud, “An intelligent cyber security phishing detection system using deep learning techniques,” *Clust. Comput.*, vol. 25, pp. 3819–3828, 2022, doi: 10.1007/s10586-022-03604-4.

[42] M. D. Abdulrahman, J. K. Alhassan, O. S. Adebayo, J. A. Ojeniyi, and M. Olalere, “Phishing attack detection based on random forest with wrapper feature selection method,” *Int. J. Inf. Process. Commun.*, vol. 7, pp. 209–224, 2019.

[43] A. K. Jain and B. B. Gupta, “Phish-safe: Url features-based phishing detection system using machine learning,” in *Cyber Security: Proceedings of CSI 2015*, pp. 467–474, 2018, doi: 10.1007/978-981-10-8536-9_44.

[44] Y. Huang, J. Qin, and W. Wen, “Phishing url detection via capsule-based neural network,” in *Proceedings of the 2019 IEEE 13th International Conference on Anti-Counterfeiting, Security, and Identification (ASID)*, pp. 22–26, 2019, doi: 10.1109/ICASID.2019.8925000.

[45] M. Zabihimayvan and D. Doran, “Fuzzy rough set feature selection to enhance phishing attack detection,” in *Proceedings of the 2019 IEEE International Conference on Fuzzy Systems (FUZZ- IEEE)*, pp. 1–6, 2019, doi: 10.1109/FUZZ-IEEE.2019.8858884.

[46] A. K. Jain and B. B. Gupta, “Detection of phishing attacks in financial and e-banking websites using link and visual similarity relation,” *Int. J. Inf. Comput. Secur.*, vol. 10, pp. 398–417, 2018, doi: 10.1504/IJICS.2018.095303.

[47] N. A. Azeez, S. Misra, I. A. Margaret, and L. Fernandez-Sanz, “Adopting automated whitelist approach for detecting phishing attacks,” *Comput. Secur.*, vol. 108, p. 102328, 2021, doi: 10.1016/j.cose.2021.102328.

[48] R. Alotaibi, I. Al-Turaiki, and F. Alakeel, “Mitigating email phishing attacks using convolutional neural networks,” in *Proceedings of the 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS)*, pp. 1–6, 2020, doi: 10.1109/ICCAIS48893.2020.9096821.

[49] S. Ruiz-Villafranca, J. Carrillo-Mondéjar, J. M. Castelo Gómez, and J. Roldán-Gómez, “Mecinot: A multi-access edge computing and industrial internet of things emulator for the modelling and study of cybersecurity threats,” *J. Supercomput.*, vol. 79, pp. 11895–11933, 2023, doi: 10.1007/s11227-023-05098-2.

[50] C. Rookard and A. Khojandi, “Applying deep reinforcement learning for detection of internet-of-things cyber attacks,” in *Proceedings of the 2023 IEEE 13th Annual Computing and Communication Workshop*

and Conference (CCWC), pp. 389–395, 2023, doi: 10.1109/CCWC57344.2023.10099349.

[51] A. Sharma, H. Babbar, and A. Sharma, "Ton-iot: Detection of attacks on internet of things in vehicular networks," in *2022 6th International Conference on Electronics, Communi- cation and Aerospace Technology*, pp. 539–545, 2022, doi: 10.1109/ICECA55336.2022.10009070.

[52] S. Subasree, N. K. Sakthivel, K. Tripathi, D. Agarwal, and A. K. Tyagi, "Combining the advantages of radiomic features based feature extraction and hyper parameters tuned rernn using loa for breast cancer classification," *Biomedical Signal Processing and Control*, vol. 72, p. 103354, 2022.

[53] S. S. Devi, A. Roy, J. Singha, S. A. Sheikh, and R. H. Laskar, "Malaria infected erythrocyte classification based on a hybrid classifier using microscopic images of thin blood smear," *Multimedia Tools and Applications*, vol. 77, pp. 631–660, 2018 DOI: 10.1007/s11042-016-4264-7.

[54] B. Bojarajulu, S. Tanwar, and T. P. Singh, "Intelligent iot-botnet attack detection model with optimized hybrid classification model," *Computers & Security*, vol. 126, p. 103064, 2023, doi: 10.1016/j.cose.2022.103064.

[55] U. G. Ketenci, T. Kurt, S. Önal, C. Erbil, S. Aktürkoˇglu, and H. Ş. İlhan, "A time-frequency based suspicious activity detection for anti-money laundering," *IEEE Access*, vol. 9, pp. 59957–59967, 2021, doi: 10.1109/ACCESS.2021.3072114.

[56] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, "Deep-cascade: Cascading 3d deep neural networks for fast anomaly detection and localization in crowded scenes," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1992–2004, 2017, doi: 10.1109/TIP.2017.2670780.

[57] C. Lee and S. Lee, "Softmax output approximation for activation memory-efficient training of attention-based networks," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[58] D. Kumar *et al*., "Grey wolf optimization based hyper-parameter optimized convolution neural network for kidney image classification," *International Journal of Intelligent Engineering & Systems*, vol. 15. No. 3, 2022, doi: 10.22266/ijies2022.0630.27.

[59] A. Alzaqebah, I. Aljarah, O. Al-Kadi, and R. Damaševičiu, "A modified grey wolf optimization algorithm for an intrusion detection system," *Mathematics*, vol. 10, no. 6, p. 999, 2022, doi: 10.3390/math10060999.

[60] B. Ahmadi, S. Younesi, O. Ceylan, and A. Ozdemir, "An advanced grey wolf optimization algorithm and its application to planning problem in smart grids," *Soft Computing*, vol. 26, no. 8, pp. 3789–3808, 2022.