Computers, Materials & Continua
DOI: 10.32604/cmc.2023.033382
Article

Probe Attack Detection Using an Improved Intrusion Detection System

Abdulaziz Almazyad, Laila Halman and Alaa Alsaeed*

Department of Computer Engineering, College of Computer Science, King Saud University, Riyadh, 11421, Saudi Arabia
*Corresponding Author: Alaa Alsaeed. Email: 442202859@student.ksu.edu.sa
Received: 15 June 2022; Accepted: 15 September 2022

Abstract: The novel Software Defined Networking (SDN) architecture potentially resolves specific challenges arising from rapid internet growth of and the static nature of conventional networks to manage organizational business requirements with distinctive features. Nevertheless, such benefits lead to a more adverse environment entailing network breakdown, systems paralysis, and online banking fraudulence and robbery. As one of the most common and dangerous threats in SDN, probe attack occurs when the attacker scans SDN devices to collect the necessary knowledge on system susceptibilities, which is then manipulated to undermine the entire system. Precision, high performance, and real-time systems prove pivotal in successful goal attainment through feature selection to minimize computation time, optimize prediction performance, and provide a holistic understanding of machine learning data. As the extension of astute machine learning algorithms into an Intrusion Detection System (IDS) through SDN has garnered much scholarly attention within the past decade, this study recommended an effective IDS under the Grey-wolf optimizer (GWO) and Light Gradient Boosting Machine (LightGBM) classifier for probe attack identification. The InSDN dataset was employed to train and test the proposed IDS, which is deemed to be a novel benchmarking dataset in SDN. The proposed IDS assessment demonstrated an optimized performance against that of peer IDSs in probe attack detection within SDN. The results revealed that the proposed IDS outperforms the state-of-the-art IDSs, as it achieved 99.8% accuracy, 99.7% recall, 99.99% precision, and 99.8% F-measure.

Keywords: GWO; IDS; InSDN; LightGBM; probe attack; SDN

1 Introduction

Advancements of Internet-based technologies constitutes a set of many networking devices with integrated circuits and electronic chips for high throughput attainment towards hardware-oriented networking. Regardless, the present infrastructure depicts specific drawbacks involving manageability, versatility, and extensibility. Network controllers and administrators are restricted to a group of pre-identified commands although it might be handy, simpler, and more effective to complement increased internet protocols and applications through network control programming in responsive and flexible ways as networking devices typically support commands and configurations following a specified embedded Operating System (OS). Additionally, scholars are bound to create their own experimental environments or incorporate simulations rather than conducting experiments on real ones for idea manifestation. By way of explanation, cutting edge and research are costly under present hardware-centric networking conditions.

The Software Defined Networking (SDN) concept was recommended with three primary layers to alleviate such shortcomings (see Fig. 1). As "an emerging network architecture where the network control is decoupled and separated from the forwarding mechanism and is directly programmable" [1]. SDN constitutes a logically-centralized controller with a network-wide view that controls many interface-configured (ForCES [2] and OpenFlow [3]) packet-forwarding devices (switches). The SDN could emerge as a novel networking advancement that unwrap current network operation and control and facilitates network advancements and novel network designs following its decoupled nature. The potential SDN advantages in current and future Internet architectures, such as information-based networking [4] has garnered much interest from the society at large.

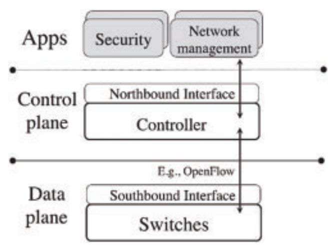


Figure 1: SDN components

Notably, SDN is exposed to probe attacks where unprotected network resources would be targeted for network damage. Following Fig. 2 [5], probe attacks attempt to gather the necessary data (IP Address, service name, operating system application, and host name) and detect network susceptibility. The attacker would employ common scanning instruments from the Internet to gather network data (nmap, satan, and mscan), which could also be utilized to instigate other attacks (Denial-of-Service (DoS), Root to Local attacks (R2L), User to Root (U2R)) beyond their essential purpose [5]. The primary idea underlying the attack originates from the perception that all rule types are only pushed from the controller to the switches, when necessary, in an SDN network. As such, a robust mechanism (automatic Intrusion Detection System (IDS)) should be provided by the network administrator for early attack detection and alleviate the risks resulting from such instances.

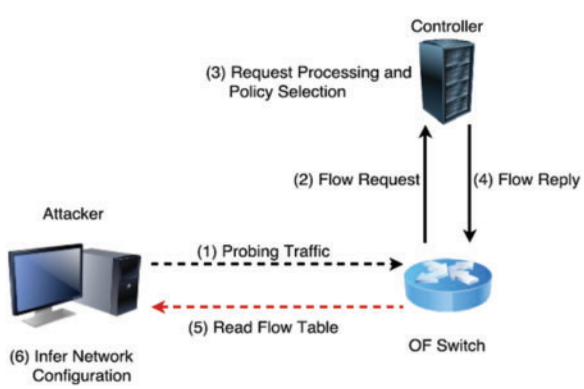


Figure 2: Probe attack scenario

The IDS is operated by monitoring and inspecting client device or network traffic behavior and serves to ascertain intrusions and suspicious activities [6]. This system issues an alarm to alert the security team and register malicious network activities into a log file for further investigation [7]. The IDS performance could be enhanced with Feature Selection (FS) to minimize computation time and intricacies through optimum feature subset selection, Microsoft proposed LightGBM in 2017 [8], a unique boosting framework that is deemed to be faster and more powerful than Xgboost [8]. The LightGBM model functioned as a classifier in the recommended IDS given its extensively acknowledged performance in resolving specific data mining and Machine Learning (ML) intricacies.

FS serves to determine a subset of features and choose the most pivotal counterpart for a classifier. As network traffic entails a substantial number of features, classifiers could yield higher precision with optimal attribute selection compared to one that is developed with a complete set of characteristics. FS could also mitigate the training dataset size given its reliable processing time and tests. Based on most empirical comparisons and demonstrations, the presence of repetitive and irrelevant features adversely affected learning model accuracy [9]. The security mechanism performance significantly relies on a subset of features chosen to be employed in optimal IDS development. As one of the extensively utilized and robust FS algorithms incorporated into various fields (IDSs), GWO selects the most crucial features that could enhance classification accuracy and intrusion detection rate.

The current study proposed an optimal IDS under GWO and the LightGBM classifier for efficient probe attack detection in SDN. The contributions of the proposed article are as follows: (i) An enhanced GWO by proposing a modified change position technique, (ii) A multi-objective fitness function to enhance performance of feature selection and classification process by selecting the most important features, and (iii) A LightGBM-based model for probe attack detection.

The remaining sections are presented as follows: Section 2 reviews relevant literature to highlight current knowledge gaps; Section 3 elaborates on the recommended IDS stages; Section 4 highlights the proposed IDS efficiency by discussing the empirical outcomes and concludes the study.

2 Related Works

Numerous attack detection methods are currently based on benchmark dataset, attack types, and simulating SDN scenarios. Robust attack detection techniques distinguish pernicious network traffic and patterns from legitimate counterparts [10]. Such techniques are extensively deployed in traditional networks and ML-assisted SDNs. For example, ML-based IDS of DDoS flooding attacks on SDNs was presented in [11]. The common principle is depicted using a case study where experimental data (jitter, throughput, and response time metrics) from a representative SDN environment, which proves adequate for typical mid-sized and enterprise-wide networks, is employed to structure classification models that precisely determine and categorize DDoS flooding attacks. The incorporated SDN model was emulated in Mininet and DDoS flooding attacks (hypertext transfer protocol or HTTP), transmission control protocol (TCP), and user datagram protocol or UDP attacks) that were launched on the SDN model with Low Orbit Ion Cannon (LOIC). On average, Classification and Regression Tree (CART) reflected the most optimal performance regarding prediction accuracy (98%), and robustness although all the examined ML techniques demonstrated high efficacy in Distributed Denial-of-Service (DDoS) flooding attack detection and classification.

A versatile modular architecture was recommended in [12] to facilitate Low-Rate Denial-of-Service (LR-DDoS) attack identification and alleviation in SDN contexts. The IDS in this study architecture was trained through six ML models. Their performance was assessed with the Canadian Institute of Cybersecurity (CIC) DoS dataset. Resultantly, the current study approach attained a 95% detection rate despite LR-DoS attack identification complexities. Regarding deployment, the open network OS controller operating on the Mininet VM employed for the simulated context for close proximity to real-world production networks. The intrusion prevention detection system alleviated all the attacks previously identified by IDS in testing topology, thus depicting the architecture utility to detect and alleviate LR-DDoS attacks.

A new DDoS attack alleviation approach in SDN-related Internet Service Provider (ISP) networks for TCP-SYN and Internet Control Message Protocol (ICMP) flood attacks employed the ML method (k-Nearest Neighbors (KNN) and Extreme Gradient Boosting (XGBoost)) following [13]. The recommended algorithms were implemented, and their accuracy evaluated to overcome the trade-off between accuracy and detection effectiveness through testbed deployment. Based on the experimental outcomes, the algorithms could effectively perform attack mitigation by over 98.0% while benign traffic proved to be unaffected. The DDoS attacks in SDN were identified with ML-oriented models parallel to [14]. Under DDoS attack traffic, particular features were first derived from SDN for the dataset in normal conditions. A novel dataset was subsequently developed with FS approaches on the present dataset for model simplification, interpretation catalyzation, and minimal training time. Both datasets that were developed with and without FS techniques were trained and tested with several ML and deep learning classifiers. Resultantly, the wrapper FS was integrated with a KNN classifier to attain the highest precision rate (98.3%) in DDoS attack identification. In this vein, ML and FS algorithms could demonstrate optimal results involving DDoS attack detection in SDN with the potential decrease of processing load and time.

Meanwhile, a learning-oriented mechanism was suggested in [15] to identify the low-rate DDoS on SDN controller and switch nodes. The proposed technique constitutes two main feature groups, namely (i) stateless group, and (ii) stateful group, elicited from the Openflow package. The IDS utilizes ML to develop classifiers and distinguish normal stream from abnormal one. The experimental environment was developed and implemented to assess the research method, which encompasses the low-rate DDoS attack module under Internet of Things (IoT) devices, the physical and virtual heterogeneous SDN network, and the data flow capture and feature extraction model. The prediction outcomes were validated from various learning algorithms and the dissemination of each raw data feature for the outcomes to be compared against conventional IP packet classification solution for the DDoS attack in IoT networks following the suggested platform. Overall, the experimental outcomes demonstrated the recommended method effectiveness.

A trigger-based IDS to detect of DDoS on data plane was recommended to detect abnormal traffic flow based on [16]. An integrated ML algorithm entailing K-Means and KNN was employed to manipulate the rate and asymmetry attributes of the flows and detect the malicious flow ascertained by the trigger-based IDS. The controller would then undertake the necessary actions to self-defend against the attacks. The recommended cooperative detection method framework involving control plane and data plane significantly enhanced detection accuracy and effectiveness and deterred DDoS attacks on SDN.

3 Proposed IDS

This section discusses the methodological stages followed to achieve the main objective of this article, namely: (i) preprocessing, (ii) GWO-based FS, and (ii) LightGBM-based attack detection.

3.1 Preprocessing

This stage strived towards data preparation for the subsequent phases (FS and detection) of the recommended IDS by converting the InSDN dataset network traffic into a more meaningful form. This stage encompasses the following components:

3.1.1 Cleansing

A significant step towards data quality and reliability reinforcement by omitting and rectifying dataset errors. Cleansing also includes managing missing, inaccurate, and noisy data that undermines model performance.

3.1.2 Transformation

Data conversion from symbolic feature values to numerical counterparts.

3.1.3 Mapping

The InSDN dataset involves specific attack types that should first be classified accordingly. As such, a mapping approach was employed to map every attack into its corresponding attack category (See Fig. 3), then each feature is indexed by integer number starting with 0, and the results is as listed in Table 1 below.

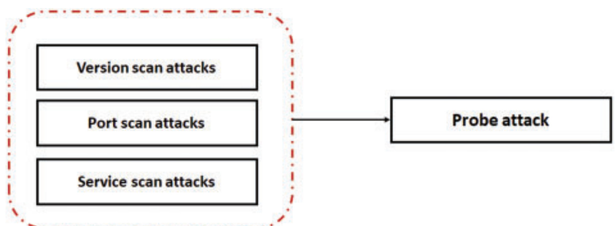


Figure 3: Attack mapping

Table 1: Features indexing

Index	Feature	Index	Feature	Index	Feature
0	Src Port	23	Fwd IAT Mean	46	PSH Flag Cnt
1	Dst Port	24	Fwd IAT Std	47	ACK Flag Cnt
2	Protocol	25	Fwd IAT Max	48	URG Flag Cnt
3	Flow Duration	26	Fwd IAT Min	49	Down/Up Ratio
4	Tot Fwd Pkts	27	Bwd IAT Tot	50	Pkt Size Avg
5	Tot Bwd Pkts	28	Bwd IAT Mean	51	Fwd Seg Size Avg
6	TotLen Fwd Pkts	29	Bwd IAT Std	52	Bwd Seg Size Avg
7	TotLen Bwd Pkts	30	Bwd IAT Max	53	Subflow Fwd Pkts
8	Fwd Pkt Len Max	31	Bwd IAT Min	54	Subflow Fwd Byts
9	Fwd Pkt Len Min	32	Bwd PSH Flags	55	Subflow Bwd Pkts
10	Fwd Pkt Len Mean	33	Bwd URG Flags	56	Subflow Bwd Byts
11	Fwd Pkt Len Std	34	Fwd Header Len	57	Init Bwd Win Byts
12	Bwd Pkt Len Max	35	Bwd Header Len	58	Fwd Act Data Pkts
13	Bwd Pkt Len Min	36	Fwd Pkts/s	59	Active Mean
14	Bwd Pkt Len Mean	37	Bwd Pkts/s	60	Active Std
15	Bwd Pkt Len Std	38	Pkt Len Min	61	Active Max
16	Flow Byts/s	39	Pkt Len Max	62	Active Min

(Continued)

Table 1: Continued

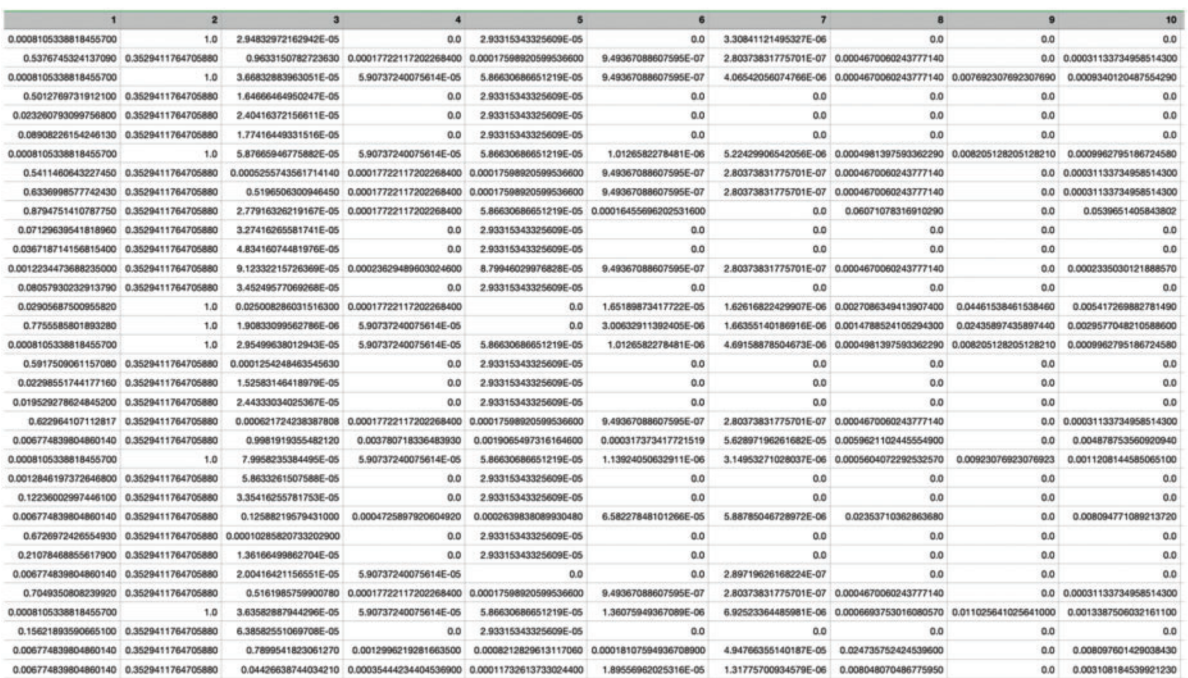
Index	Feature	Index	Feature	Index	Feature
17	Flow Pkts/s	40	Pkt Len Mean	63	Idle Mean
18	Flow IAT Mean	41	Pkt Len Std	64	Idle Std
19	Flow IAT Std	42	Pkt Len Var	65	Idle Max
20	Flow IAT Max	43	FIN Flag Cnt	66	Idle Min
21	Flow IAT Min	44	SYN Flag Cnt	67	Label
22	Fwd IAT Tot	45	RST Flag Cnt		

3.1.4 Normalization

This process denotes calibrating a range of feature values into a well-proportioned counterpart. Normalizing values range between Y_{min} and Y_{max}, which are the minimum and maximum values for feature Y with Eq. (1) and extensively utilized in recent IDS research [17].

$$Y_{new} = \frac{Y_{current} - Y_{min}}{Y_{max} - Y_{min}} \tag{1}$$

Specifically, the numerical feature values are depicted by Y. A minimal feature Y value is denoted by Y_{min} while Y_{max} demonstrates the maximum value of the same feature. The original feature Y value is indicated by $Y_{current}$, whereas the normalized feature value is denoted by X_{new}. The final dataset is as represented in Fig. 4 below.



1	2	3	4	5	6	7	8	9	10
0.0008105338818455700	1.0	2.94832972162942E-05	0.0	2.93315343325609E-05	0.0	3.30841121495327E-06	0.0	0.0	0.0
0.5376745324137090	0.3529411764705880	0.9633150782723630	0.00017722117202268400	0.00017598920599536600	9.49367088607595E-07	2.80373831775701E-07	0.0004670060243777140	0.0	0.00031133734958514300
0.0008105338818455700	1.0	3.66832883963051E-05	5.90737240075614E-05	5.86630686651219E-05	9.49367088607595E-07	4.06542056074766E-06	0.0004670060243777140	0.007692307692307690	0.0009340120487554290
0.5012769731912100	0.3529411764705880	1.64666464950247E-05	0.0	2.93315343325609E-05	0.0	0.0	0.0	0.0	0.0
0.023260793099756800	0.3529411764705880	2.40416372156611E-05	0.0	2.93315343325609E-05	0.0	0.0	0.0	0.0	0.0
0.08908226154246130	0.3529411764705880	1.77416449331516E-05	0.0	2.93315343325609E-05	0.0	0.0	0.0	0.0	0.0
0.0008105338818455700	1.0	5.87665946775882E-05	5.90737240075614E-05	5.86630686651219E-05	1.0126582278481E-06	5.22429906542056E-06	0.0004981397593362290	0.008205128205128210	0.0009962795186724580
0.5411460643227450	0.3529411764705880	0.0005255743561714140	0.00017722117202268400	0.00017598920599536600	9.49367088607595E-07	2.80373831775701E-07	0.0004670060243777140	0.0	0.00031133734958514300
0.6336998577742430	0.3529411764705880	0.5196506300946450	0.00017722117202268400	0.00017598920599536600	9.49367088607595E-07	2.80373831775701E-07	0.0004670060243777140	0.0	0.00031133734958514300
0.8794751410787750	0.3529411764705880	2.77916326219167E-05	0.00017722117202268400	5.86630686651219E-05	0.00016455696202531600	0.0	0.06071078316910290	0.0	0.0539651405843802
0.07129639541818960	0.3529411764705880	3.27416265581741E-05	0.0	2.93315343325609E-05	0.0	0.0	0.0	0.0	0.0
0.036718714156815400	0.3529411764705880	4.83416074481976E-05	0.0	2.93315343325609E-05	0.0	0.0	0.0	0.0	0.0
0.0012234473688235000	0.3529411764705880	9.12332215726369E-05	0.00023629489603024600	8.79946029976828E-05	9.49367088607595E-07	2.80373831775701E-07	0.0004670060243777140	0.0	0.0002335030121888570
0.08057930232913790	0.3529411764705880	3.45249577069268E-05	0.0	2.93315343325609E-05	0.0	0.0	0.0	0.0	0.0
0.02905687500955820	1.0	0.025008286031516300	0.00017722117202268400	0.0	1.65189873417722E-05	1.62616822429907E-06	0.0027086349413907400	0.04461538461538460	0.005417269882781490
0.7755585801893280	1.0	1.90833099562786E-06	5.90737240075614E-05	0.0	3.00632911392405E-06	1.66355140186916E-06	0.0014788524105294300	0.02435897435897440	0.0029577048210588600
0.0008105338818455700	1.0	2.95499638012943E-05	5.90737240075614E-05	5.86630686651219E-05	1.0126582278481E-06	4.69158878504673E-06	0.0004981397593362290	0.008205128205128210	0.0009962795186724580
0.5917509061157080	0.3529411764705880	0.0001254248463545630	0.0	2.93315343325609E-05	0.0	0.0	0.0	0.0	0.0
0.02298551744177160	0.3529411764705880	1.52583146418979E-05	0.0	2.93315343325609E-05	0.0	0.0	0.0	0.0	0.0
0.019529278624845200	0.3529411764705880	2.44333034025367E-05	0.0	2.93315343325609E-05	0.0	0.0	0.0	0.0	0.0
0.622964107112817	0.3529411764705880	0.000621724238387808	0.00017722117202268400	0.00017598920599536600	9.49367088607595E-07	2.80373831775701E-07	0.0004670060243777140	0.0	0.00031133734958514300
0.006774839804860140	0.3529411764705880	0.9981919355482120	0.003780718336483930	0.0019065497316164600	0.000317373417721519	5.62897196261682E-05	0.0059621102445554900	0.0	0.004878753560920940
0.0008105338818455700	1.0	7.9958235384495E-05	5.90737240075614E-05	5.86630686651219E-05	1.13924050632911E-06	3.14953271028037E-06	0.0005604072292532570	0.00923076923076923	0.0011208144585065100
0.0012846197372646800	0.3529411764705880	5.8633261507588E-05	0.0	2.93315343325609E-05	0.0	0.0	0.0	0.0	0.0
0.12236002997446100	0.3529411764705880	3.35416255781753E-05	0.0	2.93315343325609E-05	0.0	0.0	0.0	0.0	0.0
0.006774839804860140	0.3529411764705880	0.12588219579431000	0.0004725897920604920	0.0002639838089930480	6.58227848101266E-05	5.88785046728972E-06	0.02353710362863680	0.0	0.008094771089213720
0.6726972426554930	0.3529411764705880	0.00010285820733202900	0.0	2.93315343325609E-05	0.0	0.0	0.0	0.0	0.0
0.21078468855617900	0.3529411764705880	1.36166499862704E-05	0.0	2.93315343325609E-05	0.0	0.0	0.0	0.0	0.0
0.006774839804860140	0.3529411764705880	2.00416421156551E-05	5.90737240075614E-05	0.0	0.0	2.89719626168224E-07	0.0	0.0	0.0
0.7049350808239920	0.3529411764705880	0.5161985759900780	0.00017722117202268400	0.00017598920599536600	9.49367088607595E-07	2.80373831775701E-07	0.0004670060243777140	0.0	0.00031133734958514300
0.0008105338818455700	1.0	3.63582887944296E-05	5.90737240075614E-05	5.86630686651219E-05	1.36075949367089E-06	6.92523364485981E-06	0.0006693753016080570	0.011025641025641000	0.0013387506032161100
0.15621893590665100	0.3529411764705880	6.38582551069708E-05	0.0	2.93315343325609E-05	0.0	0.0	0.0	0.0	0.0
0.006774839804860140	0.3529411764705880	0.7899541823061270	0.0012996219281663500	0.0008212829613117060	0.00018107594936708900	4.94766355140187E-05	0.024735752424539600	0.0	0.008097601429038430
0.006774839804860140	0.3529411764705880	0.04426638744034210	0.00035444234404536900	0.00011732613733024400	1.89556962025316E-05	1.31775700934579E-06	0.008048070486775950	0.0	0.003108184539921230

Figure 4: Snapshot of dataset after preprocessing

3.2 *GWO-Based Feature Selection*

The GWO denotes a Swarm Intelligence Optimization algorithm inspired by the social hierarchy and hunting behavior of grey wolves. Four grey wolf types were defined to simulate the leadership hierarchy: alpha, beta, delta, and omega. The pseudocode of GWO is illustrated in Fig. 5 below.

```
Initialize the grey wolf population X_i (i = 1, 2, ..., n)
Initialize a, A, and C
Calculate the fitness of each search agent
X_α=the best search agent
X_β=the second best search agent
X_δ=the third best search agent
while (t < Max number of iterations)
    for each search agent
            Update the position of the current search agent
    end for
    Update a, A, and C
    Calculate the fitness of all search agents
    Update X_α, X_β, and X_δ
    t=t+1
end while
return X_α
```

Figure 5: Pseudocode of GWO [18]

The increased engagement of wolves in GWO would result in highly precise decisions and mitigate decision dependency. The refined GWO necessitates an additional wolf: omega wolf (ω) to reduce the impact rate of any wolf decision as thoroughly elaborated in Eqs. (2)–(9). The central updating equation is developed in Eq. (2) below [18,19]:

$$W_i^{t+1} = \text{“}Crossover\text{”}\left(w_1.\ w_2.\ w_3.\ w_4\right) \tag{2}$$

Specifically, the modified bGWO is based on this concept by adding one more wolf, called omega wolf (ω). The increase in the number of wolves that participated in the decision led to a reduction in the impact rate of any wolf's decision from 0.33% to 0.25%. Where w1, w2, w3, and w4 are binary vectors that represent the wolf move impact on alpha, beta, delta, and omega grey wolves in sequence. The w1, w2, w3, and w4 were mathematically determined in Eqs. (3)–(6), respectively.

$$w_w^d = \begin{cases} 1 & if\ \left(w_\omega^d + stepb_\omega^d\right) \geq\ 1 \\ 0 & otherwise \end{cases} \tag{3}$$

Specifically, w_ω^d denotes the location vector of the omega wolf in d while $stepb_\omega^d$ indicates a binary step in dimension d determined by Eq. (3).

$$stepb_\omega^d = \begin{cases} 1 & if\ stepc_\omega^d\ rand \\ 0 & otherwise \end{cases} \tag{4}$$

Specifically, rand implies an arbitrarily selected number from uniform distribution ∈ [0. 1] while $stepc_\omega^d$ denotes the continuous valued step size for dimension d. Eq. (5) below is employed for sigmoidal function computation:

$$stepc_\omega^d = \frac{1}{1 + e^{-10\left(A_4^d\ Di_\omega^d - 0.5\right)}} \tag{5}$$

Specifically, A_4^d, and $D_{i\omega}^d$ were mathematically determined by Eqs. (6) and (7) in dimension d, respectively.

$$A = 2b.\, r_1 - b \tag{6}$$

$$D_{i\omega}^d = |C_1.W_\alpha - W| \tag{7}$$

A simple random probability distribution crossover strategy was implemented per dimension to crossover w1, w2, w3, and w4 outcomes following Eq. (8).

$$w_d = \begin{cases} w_1^d & if\ rand < \dfrac{1}{4} \\ w_2^d & if\ \dfrac{1}{4} \le rand < \dfrac{2}{4} \\ w_3^d & if\ \dfrac{2}{4} \le rand < \dfrac{3}{4} \\ w_4^d & otherwise \end{cases} \tag{8}$$

Specifically, W1, W2, and W3 denote the weights for every objective ($\sum_1^n x_n = 1$), acc implies accuracy, miss indicates the misclassification rate, and N_{features} represents the selected number of features. On another note, TP implies true positive, TN denotes true negative, FP indicates false positive, and FN represents false negative.

Regardless, the current GWO-oriented IDS employed one objective function that induced a substantial number of utilized features, thus requiring additional network overhead, computation time, and inadequate FS. Alternatively, a multi-objective function was incorporated as a fitness function in the recommended IDS to mitigate current IDS complexities. As the study fitness assessment method, the recommended multi-objective function or weighted sum fitness function strived to minimize the number of selected features and misclassification rates and achieve high classification accuracy rates with Eq. (9). The fitness value for the recommended multi-objective function was computed with the following formula:

$$\mathrm{F}\,(\mathrm{x}) = -\left(\mathrm{v}^{*}\mathrm{accuracy} + (1-\mathrm{v})^{*}\,\frac{1}{No\_of\_features}\right) \tag{9}$$

where v is a weighting number ∊ [0, 1], accuracy denotes detection accuracy computed by Eq. (10), and No_of_features denotes the number of features selected in such iteration.

3.3 LightGBM-Based Attack Detection

As aforementioned, LightGBM is an enhanced version of the Gradient Boosting Decision Tree algorithm. The LightGBM integrates the capability of multiple decision trees in predicting/classifying classes, in order to provide the final optimal predicting/classifying generalizes. Basically, The LightGBM combines manifold “weak” learners into “strong” learners. However, there are two main causes for designing ML depending on this conception, (i) easiness in acquiring “weak” learners, and (ii) integrating more than one learner usually has superior generalization performance than utilizing one learner. Many modern studies have revealed the preponderance of LightGBM in solving many ML tasks, for instance, prediction of air quality [20] and disease detection and classification [21]. To clearly illustrate the training process of LightGBM, we take a model consisting of M trees [22], as an example described in Algorithm 1 (See Fig. 6).

Algorithm 1 The training of LightGBM.

Require: Input: Training set $\{(x_i, y_i)\}_{i=1}^{N}$

Ensure: Output: LightGBM model $\hat{y}_i^{(t)}$

Step 1. Initialize the first tree as a constant:

$$\hat{y}_i^{(0)} = f_0 = 0$$

Step 2. Train the next tree by minimizing the loss function:

$$f_t(x_i) = \arg\min_{f_t} L_{(t)} = \arg\min_{f_t} L\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right)$$

Step 3. Get the next model in an additive manner:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$$

Step 4. Repeat the Step2 and Step 3 until the model reaches the stop condition.

Step 5. Obtain and return the final model:

$$\hat{y}_i^{(t)} = \sum_{t=0}^{M-1} f_t(x_i)$$

Figure 6: LightGBM algorithm [22]

The main contribution of this article is a modified GWO that provides better performance; thus, adding one more wolf in GWO provided high performance with reduction of decision dependency. Therefore, GWO is not further vulnerable to feature selection problem. In addition, the second contribution of this article is a proposed multi-objective function, which in result leads to an appropriate selection of a subset of features.

4 Results and Discussion

This section discusses the details of benchmark dataset and evaluation metrics used to assess the performance of the proposed IDS, then, results and findings are presented in detail.

4.1 Benchmark Dataset and Evaluation Metrics

A new benchmark dataset, called InSDN [23] using Mininet simulation/SDN approaches [24,25], is utilized to assess the effectiveness of the proposed IDS. InSDN is a public attack-specific SDN dataset. It is considered the first comprehensive dataset for the SDN environment, which is used to assess the performance of IDS. InSDN contains the various attack classes that might happen in the different SDN elements. Fig. 7 illustrates the logical network topology used as a testbed to generate the InSDN dataset.

On the other hand, common evaluation metrics are used to demonstrate its performance. In order to calculate theses performance metrics, a confusion matrix is used [24,25], which is presented Table 2.

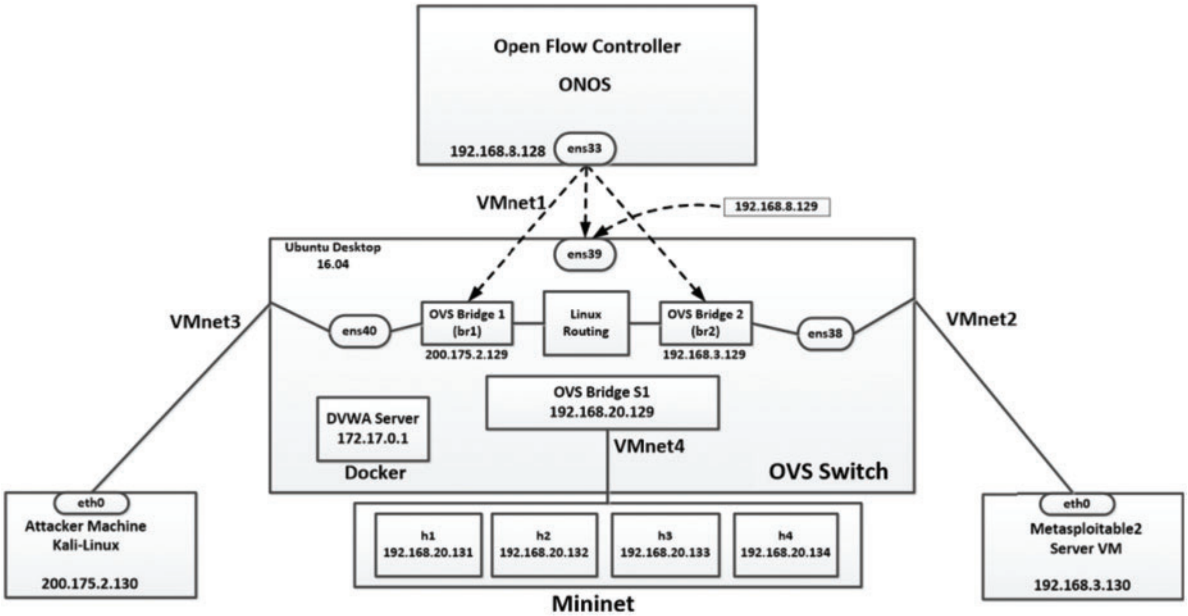


Figure 7: Logical network topology

Table 2: Confusion matrix

		Predicted	
Actual		Attack	Non-attack
	Attack	TP	TN
	Non-attack	FP	FN

The equations below are used to evaluate the accuracy, recall, precision, and F-measure [26,27], respectively of the proposed IDS:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$F - measure = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{13}$$

TP indicates the number of true positives, FN indicates the number of false negatives, TN indicates the number of true negatives, and FP indicates the number of false positives.

4.2 Experimental Setup

The proposed IDS is implemented in Python programming language. Experiments are conducted on a personal computer PC with the following hardware and software specifications, a presented in Table 3 below:

Table 3: Setup specifications

Item	Details
RAM	8 GB, DDR 4
CPU	Core i7, 10th generation
HDD	512 GB SSD
GPU	Radeon Pro 5500 XT
OS	Mac OS, OS X
Python	3.9
Configuration parameters-Mininet and OVS switch	
Hosts interfaces	Four virtual hosts (h1 to h4).
Remote controller	Four adapters in the OVS-VM, ens38, ens39, ens40, and ens41. Open flow controller ONOS.
Protocols	UDP, TCP, and ICMP.
Switch	Default OVS switch.
Link adjustment	Connect the Kali Linux VM with the same adapter of br1, and Metasploitable2 Server with the same adapter of br2.

4.3 Results and Findings

As previously mentioned, the number of features utilized in intrusion and attack detection denotes a highly crucial metric as a minimal number of features mitigates detection intricacy and time and optimizes detection accuracy and overall performance. The utilization of GWO with parameter fine tuning, as presented in Appendix Table 8, minimized the number of features from 67 to 8 after 20 runs, as depicted in Table 4. The experiments were performed with different runs to meet the requirements of computer science's test [28]. As presented in Table 5, the optimal features subset that selected contains the features with index [6 11 14 24 45 48 51 55], which are: TotLen Fwd Pkts, Fwd Pkt Len Std, Bwd Pkt Len Mean, Fwd IAT Std, RST Flag Cnt, URG Flag Cnt, Fwd Seg Size Avg, Subflow Bwd Pkts.

Table 4: Summary of FS experiments

Iteration	Best fitness	Index of selected features
0	−0.903125	[0 2 3 4 5 6 7 11 13 14 16 18 19 20 26 27 28 29 32 34 35 37 39 40 43 45 53 55 57 58 61]
1	−0.903426662	[2 3 4 7 9 11 12 13 16 18 19 20 23 24 25 26 27 32 39 40 43 45 49 55 56 58 64 65]

(Continued)

Table 4: Continued

Iteration	Best fitness	Index of selected features
2	−0.903928909	[0 2 7 9 11 12 13 16 18 19 24 25 26 27 28 34 35 43 45 48 51 55 57 64]
3	−0.904134782	[0 6 7 11 12 14 16 22 24 26 27 28 32 35 42 43 45 48 49 55 57 64]
4	−0.905263158	[0 7 11 12 13 16 19 24 26 27 32 43 45 48 51 55 59 64]
5	−0.905263158	[0 7 11 12 13 16 19 24 26 27 32 43 45 48 51 55 59 64]
6	−0.906666667	[5 6 7 11 12 14 16 19 26 27 32 49 55 64]
7	−0.908333333	[6 7 11 12 14 16 19 26 27 48 51 59]
8	−0.911111111	[6 11 12 14 16 43 45 51 55]
9	−0.911111111	[6 11 12 14 16 43 45 51 55]
10	−0.911111111	[6 11 12 14 16 43 45 51 55]
11	−0.9125	[6 11 14 24 45 48 51 55]
12	−0.9125	[6 11 14 24 45 48 51 55]
13	−0.9125	[6 11 14 24 45 48 51 55]
14	−0.9125	[6 11 14 24 45 48 51 55]
15	−0.9125	[6 11 14 24 45 48 51 55]
16	−0.9125	[6 11 14 24 45 48 51 55]
17	−0.9125	[6 11 14 24 45 48 51 55]
18	−0.9125	[6 11 14 24 45 48 51 55]
19	−0.9125	[6 11 14 24 45 48 51 55]

Table 5: Details of selected features

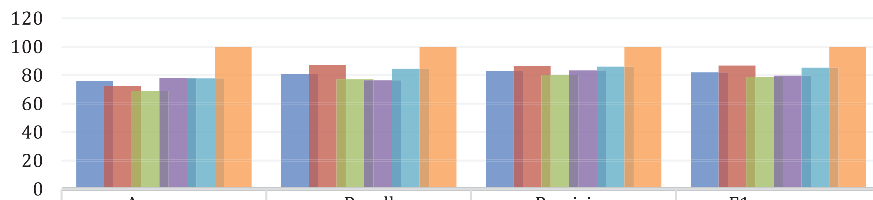
Index	Feature
6	TotLen Fwd Pkts
11	Fwd Pkt Len Std
14	Bwd Pkt Len Mean
24	Fwd IAT Std
45	RST Flag Cnt
48	URG Flag Cnt
51	Fwd Seg Size Avg
55	Subflow Bwd Pkts

The InSDN dataset with the subset of features mentioned in Table 5 is then divided into training and testing dataset, where the training dataset contains (133242) rows, and (33311) rows for testing. The LightGBM with hyperparameter, mentioned in Appendix Table 7, was trained on the training set, and then tested using the testing dataset. The experimental results obtained showed high performance, as illustrated in Table 6 below. With the use of the selected features subset, the LightGBM classifier achieved 99.8% accuracy, 99.7% recall, 99.99% precision, and f1-measure 99.8%. On the other hand, without the use of the selected features subset (i.e., with the original dataset with 67 features), the LightGBM classifier achieved 77.3% accuracy, 61.4% recall, 100% precision, and 76.1% f1-measure. These findings reveal the significant impact of using the FS (based on GWO) on enhancing the IDS performance significantly.

Table 6: Results with/without FS

Metric	Without FS	With FS
	%	
Accuracy	77.3	99.8
Recall	61.4	99.7
Precision	99.99	99.99
F1-measure	76.1	99.8

Besides, the performance of the proposed IDS was also compared against that of advanced counterparts mentioned in the literature including [11–15] to identify its efficiency. Although the IDSs attained comparable outcomes following accuracy, precision, recall, and F-measure, the proposed IDS outperformed the current IDSs in all evaluation metric as outlined in Fig. 8 below. Attaining a minimal number of pertinent network traffic elements without adversely impacting detection performance would significantly improve IDS effectiveness given the essentiality of FS in any IDS. Based on the compared methods utilizing the InSDN dataset, the proposed IDS maintains the highest performance among all state-of-the-art IDS that compared with.



	Accuracy	Recall	Precision	F1-measure
[11]	76.11	81.03	83	82.00
[12]	72.5	87.12	86.45	86.78
[13]	68.94	77.17	80.11	78.61
[14]	78.12	76.4	83.43	79.76
[15]	77.84	84.6	86.07	85.33
Proposed IDS	99.8	99.7	99.99	99.8

[11] [12] [13] [14] [15] Proposed IDS

Figure 8: Comparison with state-of-the-art IDSs

Conclusively, the proposed IDS depicted a practical means of addressing IDS complexities. The algorithm capacity to enhance the precision value and minimize the number of features for the detection process substantially optimized IDS performance. The multi-objective function (fitness function) incorporated into the fourth grey wolf explicitly affected the next algorithm position selection process. The derived experimental results reflected that the proposed IDS implied a highly positive effect on improving IDS performance compared to other current IDS methods. Although the integration of one more wolf (omega wolf or ω) with GWO offered precise decisions and decreased decision dependency, the following position in the refined GWO shifted based on the four most optimal solutions (α, β, δ, and ω) with the crossover technique. The multi-objective function also resulted in the adequate selection of a set of features that assessed whether the feature subset efficiently complemented the objectives (high detection accuracy and minimum number of features).

5 Conclusion

Intrusion detection remains one of the crucial concerns in network security. Network traffic performance is unpredictable with multiple problematic space features in the non-linear nature of intrusion attempts. The aforementioned aspects render Intrusion Detection Systems a challenge in security studies. As such, it is deemed pivotal to select essential intrusion detection components in information security. An optimal IDS method was proposed in this article following GWO and LightGBM. Several experiments were performed to reflect the proposed IDS efficiency in terms of accuracy, precision, recall and f-measures, and subsequently compared against advanced IDSs. Based on the comparison outcomes, the recommended IDS substantially optimized preliminary-stage attack detections. Given that the proposed IDS outperformed other advanced IDSs concerning accuracy, precision, recall, and F-measure, the recommended IDS proved to be more effective in preventing network attacks within SDN, especially Probe attack, compared to current sophisticated IDSs. The suggested IDS has also provided useful insights and empirical directions for anomaly identification, such as improving the next location decision by adapting the velocity parameter of the Particle Swarm Optimization algorithm.

Acknowledgement: I express my gratitude to King Saud University, The Kingdom of Saudi Arabia, for administrative and technical support.

Funding Statement: The authors would like to thank the Deanship of Scientific Research and the research Services and Support Unit (RSSU) at King Saud University for their support in this paper.

Conflicts of Interest: The author declares that they have no conflicts of interest to report regarding the present study.

References

[1] H. Farhady, H. Lee and A. Nakao, "Software-defined networking: A survey," *Computer Networks*, vol. 81, no. 11, pp. 79–95, 2015.
[2] A. Doria, R. Gopal, H. Khosravi, L. Dong, J. Salim *et al.,* "Forwarding and Control Element Separation (Forces) Protocol Specification," [Online]. Available: https://ietf.org/wg/forces/charter/.
[3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson *et al.,* "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.
[4] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher and B. Ohlman, "A survey of information-centric networking," *Communications Magazine*, vol. 50, no. 7, pp. 26–36, 2012.
[5] N. Khamphakdee, N. Benjamas and S. Saiyod, "Improving intrusion detection system based on snort rules for network probe attacks detection with association rules technique of data mining," *Journal of ICT Research & Applications*, vol. 8, no. 3, pp. 11–21, 2015.
[6] A. Alzahrani and M. Alenazi, "Designing a network intrusion detection system based on machine learning for software defined networks," *Future Internet*, vol. 1, no. 5, pp. 111–123, 2021.
[7] J. Shen and J. Wang, "Network intrusion detection by artificial immune system," in *IECON 2011-37th Annual Conf. of the IEEE Industrial Electronics Society*, Melbourne, VIC, Australia, IEEE, pp. 4716–4720, 2011.
[8] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen *et al.,* "LightGBM: A highly efficient gradient boosting decision tree," in *Neural Information Processing Systems; Neural Information Processing Systems Foundation*, vol. 30. Long Beach, CA, USA, pp. 112–135, 2017.
[9] J. Hur, S. Ihm and Y. Park, "A variable impacts measurement in random forest for mobile cloud computing," *Wireless Communications and Mobile Computing*, vol. 32, no. 11, pp. 321–339, 2017.

[10] G. Carl, R. Kesidis, R. Brooks and S. Rai, "Denial-of-service attack detection techniques," *IEEE Internet Computing*, vol. 10, no. 1, pp. 82–89, 2006.
[11] A. Sangodoyin, M. Akinsolu, P. Pillai and V. Grout, "Detection and classification of DDoS flooding attacks on software-defined networks: A case study for the application of machine learning," *IEEE Access*, vol. 9, no. 11, pp. 122495–122508, 2021.
[12] J. Perez-Diaz, I. Valdovinos, K. Choo and D. Zhu, "A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning," *IEEE Access*, vol. 8, no. 11, pp. 155859–155872, 2020.
[13] N. Tuan, P. Hung, N. Nghia, N. Tho, T. Phan *et al.,* "A DDoS attack mitigation scheme in ISP networks using machine learning based on SDN," *Electronics*, vol. 9, no. 3, pp. 413–422, 2020.
[14] H. Polat, O. Polat and A. Cetin, "Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models," *Sustainability*, vol. 12, no. 3, pp. 1035–1047, 2020.
[15] H. Cheng, J. Liu, T. Xu, B. Ren, B. Mao *et al.,* "Machine learning based low-rate DDoS attack detection for SDN enabled IoT networks," *International Journal of Sensor Networks*, vol. 34, no. 1, pp. 56–69, 2020.
[16] L. Tan, Y. Pan, L. Wu, J. Zhou, J. Jiang *et al.,* "A new framework for DDoS attack detection and defense in SDN environment," *IEEE Access*, vol. 8, no. 1, pp. 161908–161919, 2020.
[17] B. Setiawan, S. Djanali, T. Ahmad and I. Nopember, "Increasing accuracy and completeness of intrusion detection model using fusion of normalization, feature selection method and support vector machine," *International Journal of Intelligent Engineering Systems*, vol. 12, no. 4, pp. 378–389, 2019.
[18] S. Mirjalili, S. Mirjalili and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, no. 7, pp. 46–61, 2014.
[19] H. Faris, I. Aljarah, M. Al-Betar and S. Mirjalili, "Grey wolf optimizer: A review of recent variants and applications," *Neural Computing and Applications*, vol. 30, no. 2, pp. 413–435, 2018.
[20] Y. Zhang, Y. Wang, M. Gao, Q. Ma, J. Zhao *et al.,* "A predictive data feature exploration-based air quality prediction approach," *IEEE Access*, vol. 7, no. 5, pp. 30732–30743, 2019.
[21] D. Wang, Y. Zhang and Y. Zhao, "LightGBM: An effective miRNA classification method in breast cancer patients," in *Proc. of the 2017 Int. Conf. on Computational Biology and Bioinformatics*, New York NY, United States, pp. 7–11, 2017.
[22] D. Jin, Y. Lu, J. Qin, Z. Cheng and Z. Mao, "SwiftIDS: Real-time intrusion detection system based on LightGBM and parallel intrusion detection mechanism," *Computers & Security*, vol. 97, no. 1, pp. 101–117, 2020.
[23] M. Elsayed, N. Le-Khac and A. Jurcut, "InSDN: A novel SDN intrusion dataset," *IEEE Access*, vol. 8, no. 9, pp. 165263–165284, 2020.
[24] B. Lantz, B. Heller and N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks," in *Proc. of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, New York NY, United States, pp. 1–6, 2010.
[25] P. Tam, S. Math, C. Nam and S. Kim, "Adaptive resource optimized edge federated learning in real-time image sensing classifications," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, no. 1, pp. 10929–10940, 2021.
[26] A. Salih and A. Abdulazeez, "Evaluation of classification algorithms for intrusion detection system: A review," *Journal of Soft Computing and Data Mining*, vol. 2, no. 1, pp. 31–40, 2021.
[27] S. Ludwig, "Applying a neural network ensemble to intrusion detection," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, no. 1, pp. 11–25, 2019.
[28] J. Devore, "Probability and Statistics for Engineering and the Sciences," *Cengage Learning*. [Online]. Available: https://fac.ksu.edu.sa/sites/default/files/probability_and_statistics_for_engineering_and_the_sciences.pdf.

Appendix Hyperparameters

Table 7: LightGBM hyperparameters

learning_rate	0.05744
num_leaves	8
max_bin	380
bagging_freq	5
bagging_fraction	0.7003
feature_fraction	0.4800
lambda_l1	2.5
lambda_l2	4.5
min_child_samples	3
bagging_seed	42
metric	auc
random_state	451
max_drop	50

Table 8: GWO parameters

Max_iter	20
SearchAgents_no	68
lb lower limit	0
ub upper limit	1