EAI Endorsed Transactions
on Pervasive Health and Technology
Research Article EAI.EU

# Hybrid Detection and Mitigation of DNS Protocol MITM attack based on Firefly algorithm with Elliptical Curve Cryptography

Sabitha Banu. A.[1,*], Dr. G. Padmavathi[2]

[1] Department of Computer Science, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, India.

[2] Department of Computer Science, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, India. Email: padmavathi.avinashilingam@gmail.com

## Abstract

A Domain Name Server is a critical Internet component. It enables users to surf the web and send emails. DNS is a database used by millions of computers to determine which address best answers a user's query. DNS is an unencrypted protocol that may be exploited in numerous ways. The most popular DNS MITM attack uses DNS poisoning to intercept communications and fake them. DNS servers do not verify the IP addresses they forward traffic to. In DNS attacks, the attacker either targets the domain name servers or attempts to exploit system weaknesses. The Proposed FFOBLA-ECC model detects the DNS Spoofed nodes in a wireless network using the optimized firefly boosted LSTM with the help of TTL and RTR parameters received from the simulation environment and provides authentication between the nodes in order to mitigate it using the Elliptical curve cryptography. The proposed model results are different from the other methods and yield highly accurate results beyond 98% compared with the existing RF, ARF, and KNN methods.

**Keywords:** Domain Name Service(DNS), Man in the Middle attack(MITM), DNS MITM attack, Firefly algorithm, Elliptical Curve Cryptography(ECC).

Received on 29 March 2022, accepted on 16 August 2022, published on 25 August 2022



doi: 10.4108/eetpht.v8i4.3081

## 1. Introduction

The Internet is a haven for all kinds of cyber-attacks. For protection against cyber-attacks, many internet platforms and services designed to make lives more accessible at the same time must be aware and vigilant. Some attacks on web servers and websites have grown simpler due to the Internet's wide accessibility, which causes risks that lead to revenue loss in organizations. One of the most significant risks associated with the cyber-attack is based on the "Domain Name System (DNS)" protocol, which acts as a fundamental internet protocol, DNS Spoofing or DNS MITM (Man in the Middle) attack [1]. The domain name to IP address mapping is made easier with this hierarchical, distributed directory service. DNS Spoofing or DNS MITM attack is where the hackers compromise the system, inject false DNS records in the DNS servers, and redirect the traffic to some fraudulent websites to steal the credentials or information.

As soon as a domain name is entered into a web browser, the DNS resolver requests to translate the domain name to an IP address. If the domain name isn't discovered in the DNS resolver, then something went wrong. If that's the case, it'll try to find the data on another recursive or iterative server and pass it along to the user so they may follow it. In the DNS server's memory, requests for domain addresses are stored for later use.

DNS is a critical name resolution service that is extensible to clients [2]. However, owing to the enormous increase of internet users, DNS became insecure. Attackers actively seek ways to exploit the system's flaws. DNS cache poisoning, spoofing, or MITM, is a misleading cyber-attack that changes DNS

* Corresponding author. Email: sabithabanu_cs@yahoo.com

records via a DNS query that redirects traffic from legitimate servers to false websites. These attacks are difficult to detect and sometimes undetected for a long time, causing serious security issues.

DNS spoofing attacks are classified according to the attacker's ultimate objective. DNS spoofing may be accomplished in a variety of methods, including the following [3]:

- Compromising a DNS server
- Man in the Middle attack
- DNS Cache Poisoning attack
- Creating False base station network and fabricate the DNS server.

The Global DNS threat report (2019) [4] shows that 82 percent of companies are impacted, with 63 percent suffering outages. The research also showed that financial services companies pay the most per DNS attack compared to other industries.

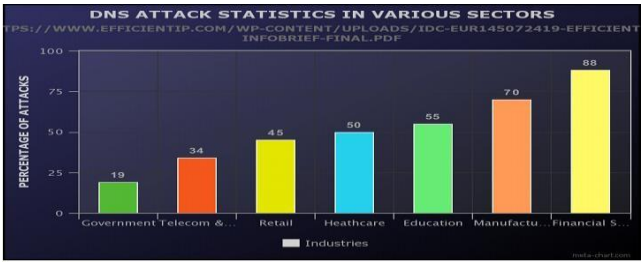



Fig 1 Percentage of DNS Attacks on Various Industries

An example of how DNS Spoofing or DNS MITM works is shown in Fig 2.

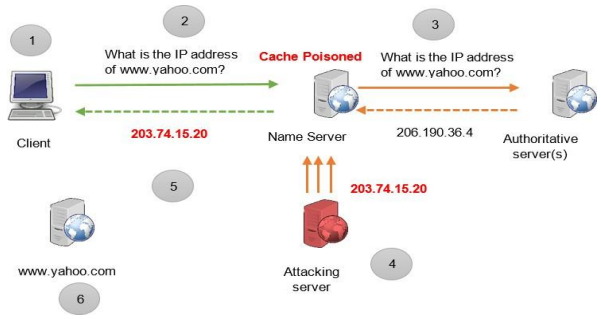



Fig 2 How DNS Spoofing or DNS MITM works.

Different types of DNS attacks have been studied. Among those, DNS MITM or DNS Spoofing are the most predominant and commonly found on the Internet. DNS MITM allows the attackers for data theft, malware infection, stop the security updates and censorship.

The main objective is to predict the detection of the DNS Spoofed nodes in a network using the FFOBLA-ECC algorithm based on the simulation data analysis. The proposed approach creates a network with a server, clients (browsers), and a controller to generate traffic between them. The firefly algorithm optimizes the characteristics based on the generated traffic. The characteristics are given as input into the LSTM model to stabilize it, and then AdaBoost is used to enhance the detection accuracy and provides authentication of the nodes.

The secondary objectives are as follows:

(i) The Proposed FFOBLA-ECC detection model detected the DNS Spoofed nodes in a wireless environment based on the simulation data without modifying the protocol.
(ii) The predicted accuracy based on the simulation model outperformed the existing Random Forest, Adaptive Random Forest, and KNN by a success rate of 99%.

The rest of the article is structured as follows. Background studies have been discussed in the next part, The FFOBLA-ECC model in part 3 with the detailed protocol and simulation approach. The technical experiment details with the outcomes are discussed in part 4, and Part 5 ends with the final part of the paper.

## 2. Background Study

Some of the necessary online tools that are in practice to handle DNS MITM attacks are

- Ettercap
- Hetty
- Bettercap
- Proxy.py
- Mitmproxy and
- Burp

Based on the literature, different types of Detection mechanisms, Prevention mechanisms, Defense mechanisms, Cryptographic mechanisms, and ML/DL mechanisms to handle the DNS MITM attacks are given in Tables 1,2,3,4 and 5.

The following table 1 summarizes the literature on the various kinds of DNS MITM attack detection methods.

Table 1 DNS MITM attacks Detection mechanisms

| Authors | Journal, issue, Year | Detection Mechanism | Limitations |
|---|---|---|---|
| M.Antonakakis et al. [5] | Springer, 2010 | Anax(Real Time) | Low rate of False Positives Increased Detection Rate |
| Yong Wan Ju et al. [6] | IEEE, 2007 | Cache Poisoning Detection System(CPDS)( Real Time) | Lower dependency, higher safety, applicability. |
| Park et al.[7] | ACM Proceedings, 2004, Vol 6 | CoDNS | Less secured |

| | | | |
|---|---|---|---|
| L.Poole [8] | ACM Proceedings, 2006, Vol 3 | ConfiDNS | Improved security |
| Sun et al.[9] | Springer, 2009 | DepenDNS Protocol | Efficient |
| Nadhem J. AlFardan, Kenneth G. Paterson [10] | Springer Proceedings,2010 | DepenDNS Protocol | DNS cache poisoning and DDoS attacks went undetected. |
| L.Yuan et al. [11] | IEEE Proc, 2006, Vol 5 | Dox | Simple and modest, improve the coherence of DNS Caches |
| Roberto Perdisci et al. [12] | Proc, 2008 | WSEC DNS (Wild-card Secure DNS) | Overhead in terms of CPU and memory consumption. |
| Roberto Perdisci et al. [13] | IEEE Proc, 2009 | WSEC DNS(Wild-card Secure DNS) | - |
| B.B.Jaya Singh[14] | CVR Journal of Science and Tech,2017, Vol 12 | Detection Algorithm | - |
| Sergii Lysenko et al. [15] | Proc, 2020 | Isolation Forest algorithm | Blocks the malicious domains based on the DNS traffic |
| Pratik Satam et al. [16] | Journal of Internet Services and Information Security, 2015, vol no.4 | DNS-IDS | |
| Chirag Sharma [17] | Thesis, 2020 | Feed-Forward Neural Network machine learning model | Not applied on the real-time dataset. |
| Caiyun Huang[18] | IEEE Proc, 2019 | Self-Feedback Detection System(SFDS) | High accuracy, effective |
| Yasuo Musashi et al. [19] | IPSJ SIG Technical Report,2011, Vol 53 No 1 | Damerau-Lavanshtein Distance based detection model | High accuracy |
| Yong Wan Ju et al. [20] | IEEE Proc, 2007 | CPDS: Cache Poisoning Detection System | Improved trustability |
| Artem A. Maksutov [21] | IEEE Proc, 2017 | DNSwitch | Applied only on LAN |
| NM Sahri et al. [22] | IEEE Proc, 2016 | CAuth | Less computation time, high bandwidth, lightweight method |
| Hao Wu et al. [23] | IEEE Proc, 2015 | Kalman filter | Quick and effective |

The following table 2 summarizes the literature on the various kinds of DNS MITM attack Prevention mechanisms.

Table 2 DNS MITM attacks Prevention mechanisms

| **Authors** | **Journal, issue, Year** | **Prevention Mechanisms** | **Limitations** |
|---|---|---|---|
| Nabih Abdelmajid et al. [24] | Proc, 2020 | GPS technology for authentication | Drawbacks in Fixed location |
| Ramzi Bassil et al. [25] | IEEE Proc, 2012 | S-DNS | Backward compatible, simple, low communication overhead, low computation overhead, efficient encryption key management scheme. |
| Jin Cao et al. [26] | Springer, 2017, | Selective Re-Query Case Sensitive Encoding scheme | Increased security, network efficiency. |
| Haider Salim Hmood et al. [27] | Security in Computer Systems and Networks The Computer Journal, 2015, Vol. 58 No. 4 | Adaptive-Cache of DNS (ACDNS) | Accurate enough, reduce latency |

| | | | |
|---|---|---|---|
| Lejun Fan et al. [28] | IEEE Proc, 2011 | Security Proxy | Easy to implement, deploy, efficient |
| Yan Zhao et al. [29] | (JISIS), 2020, vol 10, no 2 | DCG client-side protection module | compatible with CDN mechanism, pluggable service |
| Jayashree Mohan et al. [30] | IEEE Proc, 2015 | Bi-Query and S-Key based encoding scheme | Reduces the probability of attack rates |
| Talha Naqash et al. [31] | IEEE Proc, 2012 | Security proxy with SHA-1 | Simple to execute |
| Shimrit Tzur-David et al. [32] | Springer, 2011 | Delay Fast Packets | Less memory consumption, scalable, not able to detect DoS attack |

The following table 3 summarizes the literature on the various kinds of DNS MITM attack defence mechanisms.

Table 3 DNS MITM attacks Defence mechanisms

| Authors | Journal, issue, Year | Defense Mechanisms | Benefits |
|---|---|---|---|
| Zheng Wang et al. [33] | Springer Lec.Notes, 2017, Vol 238 | On-Demand Defense (ODD) | Reduces DNSSEC overhead, improves the performance |
| Tengchao Ma et al. [34] | IEEE Proc,2020 | Adaptive dynamic Defense Strategy | Converges to optimization at the local level as opposed to global |
| Sze Yiu Chau et al. [35] | Springer,2018, Vol 255 | CGuard | Easily deployable, compatible |
| Jonathan Trostle et al. [36] | IEEE Proc,2010 | DNS Proxy Server | Minimum performance |

The following table 4 summarizes the literature on the various kinds of DNS MITM attack cryptographic methods.

Table 4 DNS MITM attacks Cryptographic mechanisms

| Authors | Journal, | Cryptographic | Limitations |
|---|---|---|---|
| P.Vixie et al.[37] | IETF, 2000 | TSIG | Low-cost to calculate keyed-hash authentication codes |
| D.E.Eastlake et al.[38] | IETF, 2000 | SIG(0) | Expensive asymmetric operations |
| R.Arends et al. [39] | IETF, 2005 | DNSSEC | Do not provide buffer overruns; DoS attacks confidentiality. It can't be deployed in a short period. Increases workload traffic. |
| D.J.Bernstein [40] | Online, 2009 | DNSCurve | No end-to-end security, DNS response validity is not notified. |
| H.M.Sun et al. [41] | Springer, 2009 | S-DNS | Time to Live (TTL) and authoritative answer policies are not adopted. The users accept Outdated SSL certificates and bad certificates. |
| Naveen Kumar & Kamal Kumar Ranga[42] | IJIRST, 2015, Vol 2, Issue 1 | elliptic curve digital signature algorithm | More minor keys with high security, low power consumption. |
| Manisha Singh & Snigdha [43] | Easy chair, 2020, Preprint no.3793 | Message Digest algorithm and RNG(Pseudo Random Number Generator) algorithm, DSA algorithm | |
| Mohammed Abdulridha | IEEE Proc, 2016 | asymmetric cipher | delay |

| | | | |
|---|---|---|---|
| Hussain et al. [44] | | | |
| M. H. Jalalzai [45] | IEEE Proc, 2015 | cryptographic framework | |
| Xue Jun Li et al. [46] | IEEE Proc, 2019 | character-wise encryption (CWE) method | Reduced encryption time |

The following table summarizes the literature of ML/DL techniques to handle DNS MITM attack.

Table 5 Literature of ML/DL Techniques used for DNS MITM attack

| Author | Journal, issue, Year | ML /DL Techniques | Limitations |
|---|---|---|---|
| Bai et al.[47] | Springer, 2011 | ANN | Accuracy detection-98% (valid and forged packets) |
| Liguo chen et al.[48] | Elsevier Proceedings, 2018, Vol 134 | Random Forest Algorithm | Classification model FPR-0.0% FNR-4.36% |
| Mykola Kozlenko, Valerii Tkachuk [49] | Proceedings, 2019 | RNN | Accuracy detection-70% |
| Abdallah Moubayed et al. [50] | IEEE Proc, 2020 | ensemble-based feature selection and bagging classification model | Lower computational complexity |
| Cho Do Xuan et al. [51] | IJETER,2020, Vol 8, No 5 | Random Forest | Time-consuming on extracting features |
| Yong Jin et al. [52] | IEEE Proc, 2019 | SVM | Not tested in a real-time network. |

## 2.1. Observation due to literature

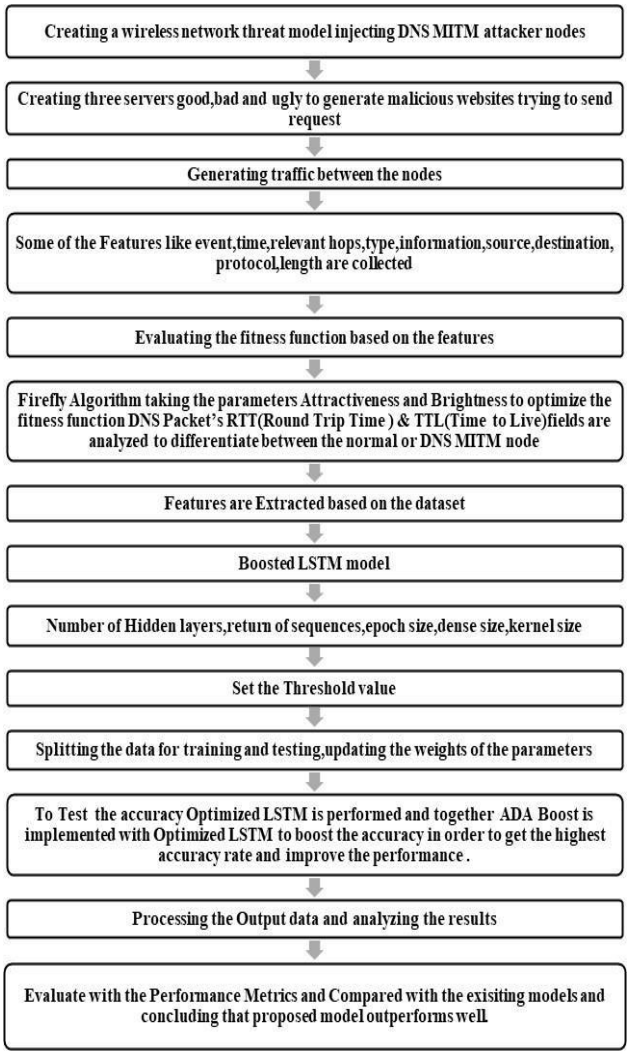
Among all this literature, ML/DL Techniques have given a high accuracy rate where some methods have reduced time in extracting features, some reduced computation time. However, researchers have rare literature on using swarm intelligence algorithms defined in the proposed methodology.

# 3. The Proposed Model

The Proposed FFOBLA-ECC model consists of a server, clients, and a controller which generated the communication between the server and the clients. The model predicts DNS MITM attacker nodes based on the optimized firefly boosted LSTM method, and also it is effective in terms of accurate detection [53].

## 3.1. DNS MITM attacker model

A server, clients (browsers), and a controller are included in the proposed attacker model. The controller generates traffic between the clients and servers. The obtained traffic is captured as an a.csv file, and the firefly algorithm is used to optimize the features. The features are loaded into the LSTM model to stable the model, and the network is then stimulated using AdaBoost to optimize detection accuracy




## 3.2 Protocol Simulation model

An HTTP toolkit with 50 browser clients, three servers, and a single controller is utilized to handle and run the

simulation. In the simulation, the client and server use the TCP/IP model to send messages. Clients and servers generate requests and replies[54]. The scenario includes clients, servers, and two routers linked by a connection. The servers are www.good.com, www.bad.com, and www.ugly.com. The network structure defines the message that may be sent directly between clients and servers or through routers. Some of the browser-based parameters evaluated for the request and reply are given in Table 6.

Table 6. Browser-based Parameters

| Parameters |
|---|
| sessionInterval |
| requestInterval |
| reqInSession |
| processingDelay |
| requestSize |
| pageSize |
| numResources |
| textImageResourceRatio |
| imageResourceSize |
| textResourceSize |
| replyDelay |

As part of the attack, the attackers compromised code in the victim's web browsers, diverting traffic to the attacker's servers and spoofing the victim's IP address. As demands increase, so does the server service level. On average, per 200 queries, the browser user receives one malicious link. Even if discovered, the user is routed to the attacker's site. The firefly optimization method collects browser request and response data over time to identify DNS MITM attacks.

## 3.3. Optimized firefly boosted LSTM Algorithm

The simulation generates datasets in the form of a .csv file. Training and testing datasets are separated. The flashing characteristic of fireflies is the basis for the firefly algorithm [55]. The three rules that apply to flashing features are as follows:

- Fireflies are genderless, so that they may be attracted to each other despite their differences in appearance.
- They are also attractive based on their brilliance; thus, the brighter one attracts the less vivid one.
- The target feature's search space to be optimized affects the brightness or intensity of a firefly's light.

Based on the firefly's brightness, the objective function is optimized on how the brighter firefly is attracted to the other. At the same time, the distance between fireflies enhances attraction. Fireflies move spontaneously if their brightness is the same. A novel solution created by chance and firefly attraction. Two key issues are at stake: **light intensity variance** and **attractiveness formulation**. A firefly's brightness $I$ at a specific location x defined as $I(x)$, and it is proportional to the fitness function F I(x)αF(x).

The brightness of the Firefly I(r) changes with respect to the distance r may be defined as

$$I = I_0 e^{-\gamma r} \quad (1)$$

Where $I_0$ is the brightness, and $\gamma$ is the coefficient of absorption of light. The attractiveness $\beta$ is equivalent to the brightness of the Firefly and brightness $\beta$ can be defined as

$$\beta = \beta_0 e^{-\gamma r^2} \quad (2)$$

Where $\beta_0$ defines the attractiveness at r=0. Brightness I and attractiveness $\beta$ are linked in several ways. Although brightness is a true measure of the firefly's emitted light, attractiveness is a subjective measure of the light that should be seen in beholders' eyes and measured by other fireflies. Distance ($r_{ij}$) between the fireflies $i$ *and* $j$ defined using Cartesian distance

$$r_{ij} = \| x_i - x_j \| = \sqrt{\sum_{k=1}^{k=n} (x_{ik} - x_{jk})^2} \quad (3)$$

N defines the dimensionality of the problem. The firefly $i$ migrate towards the brightness of the other firefly $j$ is defined as

$$x_i = x_i + \beta_0 e^{-\gamma r^2} (x_j - x_i) + \alpha \varepsilon_i \quad (4)$$

Where $\varepsilon_i$ is a random number that denotes the movement of the fireflies based on the Gaussian distribution. The fireflies' movement comprises the firefly's current position, attractiveness towards the other firefly, and random walk a generated randomly from the interval (0, 1).$\gamma$ denotes the convergence speed. The value is taken from $\gamma \epsilon (0, \infty)$. $\beta_0$ denotes the random walk.

The Firefly method uses the DNS header file's TTL and roundtrip time to identify DNS MITM attacks. Other features retrieved from the simulation environment include Event, Time, Router, ID, Source got, Source received, Actual received, Actual utilized. Extracted features put into LSTM model to address vanishing gradient. Another kind of recurrent neural network used for time series data prediction LSTM, has several faults, including vanishing gradient issues. It cannot train the model for long-term dependency. LSTM solves these problems by storing data for an extended period utilizing these three gates.

LSTM model uses three gates called input gate, forget gate, and output gate. Each gate learns from the input given through a chain of sequences and chooses whether to retain or reject it to transmit necessary information down the lengthy chain of sequences. The equation of input ($i_t$), output ($o_t$)and forget ($f_t$) states are

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \quad (5)$$

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \quad (6)$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \quad (7)$$

$$\tilde{c}_t = \tanh(w_c[h_{t-1}, x_t] + b_c \quad (8)$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \quad (9)$$

$$h_t = o_t * \tanh c_t \quad (10)$$

Where $\sigma$ denotes the sigmoid function, $w_x$ denotes the weight of the respective neuron, $h_{t-1}$ denotes the output received from the earlier iteration at the checksum *t-1*, $x_t$ denotes entering data of the current checksum, $b_x$ denotes corresponding gates' prejudices, $c_t$ denotes cell state memory at the checksum (t), $\tilde{c}_t$ denotes a possible option for the cell state at the checksum (t). The input gate symbolizes the new information that will be saved in the cell at the moment. The output gate activates the last LSTM block using a sigmoid function, checksum 't' received from the forget state. Forget state represents deleted information. To find a candidate ($c_t$) for the current checksum (t), and from the equations above, the model knows which has to be discarded i.e., $f_t * c_{t-1}$ and which has to be taken for the current checksum (t) i.e., $i_t * \tilde{c}_t$. Then the cell state is filtered and passed t the activation function to predict the information that appears in the LSTM block's output in the current checksum (t). $h_t$ is passed to the current LSTM block with the softmax layer to predict the output $y_t$.

When LSTM is trained, the model's performance is continuously unstable. To overcome it, one of the robust ensemble machine learning techniques called AdaBoost, which is good at predicting results, is ensembled with an LSTM block to train iteratively after t iterations.

$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \quad (11)$$

where

$$\alpha_t = \frac{1}{2}\log \frac{1-s_t}{s_t} \quad (12)$$

$\alpha_t$ denotes the weights of the weak learners in the classifier and combines all the t predictors to boost the model's performance to achieve high accuracy, which results in a stable form. Both AdaBoost and LSTM models are trained separately, and predictions are calculated on average. This combined heterogeneous model yields accurate prediction results than using a single LSTM model.

This hybridized optimized firefly with boosted LSTM gives high accuracy, which outperforms the existing Random Forest, Adaptive Random Forest, KNN models in terms of some performance metrics like Detection Accuracy, Delay, Packet Delivery Ratio, Packet Drop Ratio, Throughput.

**Pseudo Code**

```
1. Identifying Objective function f(x), x=(x1, x2, ... , xd)T
2. Initialize a population of fireflies xi (i = 1, 2, ... , n)
3. Define light absorption coefficient gamma
4. WHILE count < MaximumGenerations
5. FOR i = 1 : n (all n fireflies)
6. FOR j = 1 : i
7. Light intensity Ii at xi is determined by f(xi)
8. IF Ii > Ij
9. Move firefly i towards j in all d dimensions
10. ELSE
11. Move firefly i randomly
12. END IF
13. Attractiveness changes with distance r via exp[-γ r2]
14. Determine new solutions and revise light intensity
15. END FOR j
16. END FOR i
17. Rank the fireflies according to light intensity and find the current best
18. END WHILE
19.Define Features_Cols(Event,Time,Router ID,Source got,Source
Received,Actual Received,Actual Used)
20. Assign x=feature_Cols ,y=label
21. Define X_train,X_test,y_train,y_test
22. Define num_words = 1150892
23. Define LSTM model = Sequential()
24. Set LSTMmodel(Embedding(num_words, 6, input_length=8))
25. Set LSTMmodel.add(Dropout(0.1))
26. Set LSTM model(25, dropout=0.1, recurrent_dropout=0.1))
26 Set LSTM model(Dense(75, activation='relu'))
27. Set LSTMmodel(Dropout(0.1))
28. Set LSTMmodel.add(Dense(1, activation='sigmoid'))
29.Set LSTMmodel.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
30. Set LSTMmodel.fit(X_train, y_train, batch_size=54, epochs=10)
31.Print Detection Accuracy.
32.Stop
```

## 4. Experiment Results And Discussions

The Technical specifications of the experiment conducted in a simulated environment are given in table 7. Six months of data logs were taken for the experiment.

The Proposed hybrid FFOBLA-ECC model gives a novel solution to detect and mitigate the DNS MITM attack nodes using the Optimized firefly boosted LSTM and ECC algorithm. It produced very accurate outcomes when compared with the existing RF, ARF, and KNN models. The advantages of this proposed hybrid FFOBLA-ECC model include its ease of convergence into complicated problems, its simplicity and versatility, and its accurate, high-performance outcomes.

Table 7. Simulation Parameters

| Parameters | Values |
|---|---|
| Number of Access requests | 50 |
| Initial energy level | 100(j) |
| Number of Objects | 50 |
| Energy consumption for reception and transmission | .395(w),660(w) |
| Zone of simulation | 1500m×1500m |
| Transmission Range | 100kbps |
| Number of Services | 50 |
| Simulation time | 3600s |

Simulation is done, and the data are collected, and the packet details are

- Event
- Time
- Relevant Hops
- Type
- Information
- Source

- Destination
- Protocol
- Length

Performance measures are used to compare the Existing and Proposed approaches. Additionally, they are

- Detection Accuracy
- Delay
- Packet Delivery Ratio
- Packet Drop Ratio
- Throughput.

## 4.1. Detection Accuracy Ratio

The proportion of correctly predicted DNS MITM nodes is about the total number of predictions produced. It can be shown from the detection accuracy ratio that the hybrid FFOBLA-ECC model outperforms the current methods.

$$DR = \frac{TP + TN}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

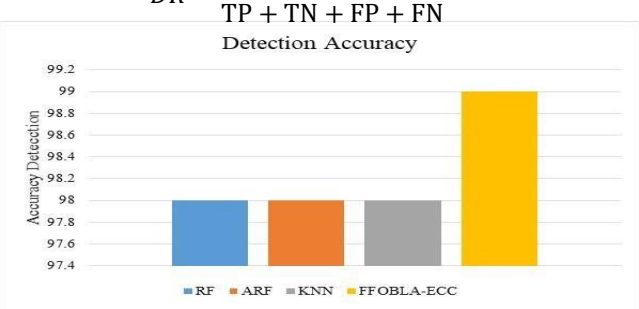



Fig 3 Accuracy Detection Ratio for DNS MITM Nodes

The given graph demonstrates that the proposed hybrid model accurately computed the DNS MITM attack nodes when applied to the simulation dataset. Existing techniques such as Random Forest, Adaptive Random Forest, and KNN have shown detection accuracy of 98%, 98%, and 98%, respectively. The Proposed hybrid model, compared to other methods, the optimized firefly boosted LSTM algorithm outperforms them all and obtained a detection accuracy of 99%.

## 4.2. Average Delay

The time it takes for data packets to arrive at their destination. Furthermore, it takes time to find the routes and queue packets for transmission. It only counts packets that are delivered to their destination. When the average Delay outcomes are low, it indicates better performance. Average delay is shown in Fig 4.

$$Average\ Delay = \frac{\text{Total no of arriving time} - \text{Total no of sending time}}{\text{Total no of Packets}}$$

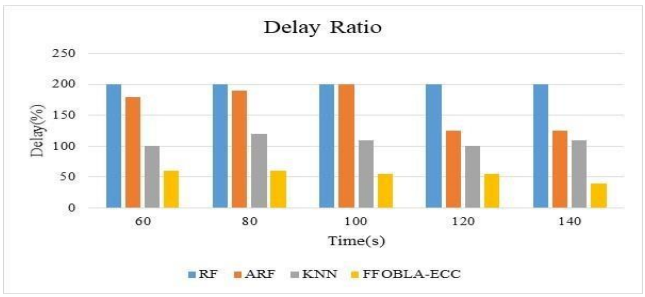



Fig 4 Average Delay

## 4.3. Packet Delivery Ratio

The source node's and the destination node's combined packet counts. The mathematical equation used to calculate PDR is shown below.

*Packet Delivery Ratio= (Packets Received/Packets Sent) x 100%*

Packet Delivery Ratio Fig 5 is given below.

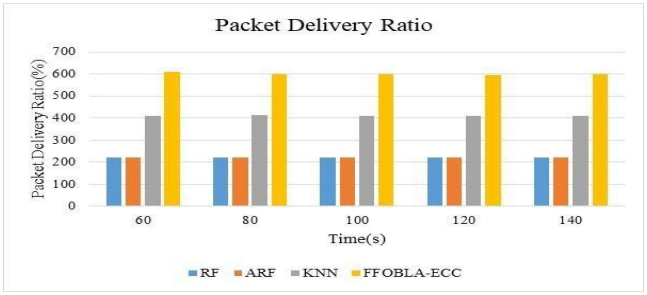



Fig. 5 Packet Delivery ratio

## 4.4. Packet Drop Ratio

Packet loss is a percentage of overall packet transmission time. It is calculated by

$$\text{Packet drop ratio}(\%) = \text{sent packets} - \text{received packets}$$

Packets drop ratio Fig 6 is given below.

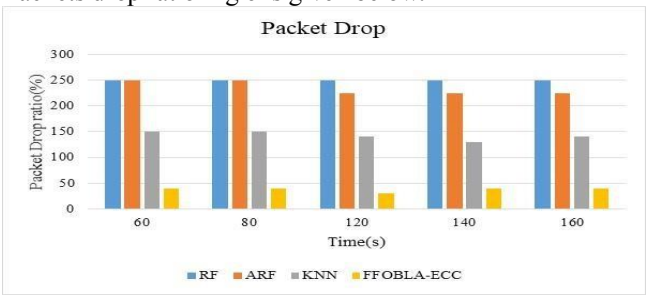



Fig 6 Packet drop Ratio

## 4.5. Throughput

Throughput is the number of packets successfully acquired in a given period, measured in bits per second (bps).Throughput Fig 7 given below

$$thruput = \sum \frac{(Traffic\ received - Traffic\ sent)}{Total\ data\ packets\ received}$$

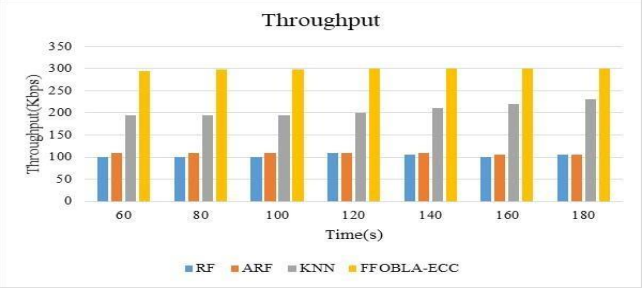



Fig 7 Throughput

Table 7.7 compares the proposed FFOBLA-ECC model to existing methods in terms of performance measures. The proposed hybrid FFOBLA-ECC Method performed optimally based on time (s) basis as well as the total number of nodes, resulting in a small percentage of delay. Packet Delivery Ratio was the most outstanding level reached 600 percent compared to previous techniques that attained just 200 percent, resulting in packet delivery success. The packet drop percentage is low, ranging between 30% and 40%. The current techniques RF, ARF, and KNN, had the most significant drop ratios ranging from 150 to 250 percent, but the maximum throughput ratio of 300 percent remained constant. All of the time and current techniques range from 100 percent to 225 percent, resulting in the best packet transmission.

Table 7 Comparison of Existing Model and Proposed Model

| Metrics used for comparison | Existing: RF (bps) | Existing: ARF (bps) | Existing: KNN (bps) | Proposed: FFOBLA-ECC (bps) |
|---|---|---|---|---|
| Detection Accuracy | 98 | 98 | 98 | 99 |
| Average Delay | 200 | 200 | 100 | 60 |
| Packet Delivery Ratio | 220 | 220 | 410 | 610 |
| Packet Drop Ratio | 250 | 250 | 150 | 40 |
| Throughput | 100 | 110 | 195 | 295 |

## 5. Conclusion

The DNS MITM attack is where the attacker holds the DNS records and tries to change them so that it can redirect the traffic to fake websites to steal the credentials of the victims or access some sensitive information or malicious websites try to install worms or virus software in their personal computer for long term access to the data stored in the computer.

DNS MITM attack can be troublesome for both the website owners and the users. It can cause some security issues and be undetected for an extended period. There are various prevention techniques applied to avoid DNS MITM attacks using some encryption techniques, DNSSEC. Various literature was given to detect, prevent, and defend against the DNS MITM attack. The Proposed hybrid Model uses a threat model of creating a DNS MITM attack using a simulation tool and injecting the attack, capturing the traffic before and after the attack.

Using Optimized Fireflies boosted LSTM algorithm provides a novel solution in detecting and predicting the attacker nodes' accuracy by using the captured file obtained from the simulation. Based on the TTL and Roundtrip time of the DNS record, some features are extracted and fed into the LSTM model for the training. It uses AdaBoost with the LSTM model to stabilize the network and improve its performance to achieve high accuracy results. Boosted LSTM performs well than the single LSTM model.

The Proposed FFOBLA-ECC Model outperforms the existing model and yields a high accuracy rate with performance.

## References

[1] https://www.cloudflare.com/en-in/learning/dns/what-is-dns/

[2] Sinéad Hanley, *DNS Overview with a discussion of DNS Spoofing*, 2000

[3] Sehgal A., Dixit A, *Securing Web Access—DNS Threats and Remedies,* In: Rathore V., Worring M., Mishra D., Joshi A., Maheshwari S. (eds) Emerging Trends in Expert Applications and Security. Advances in Intelligent Systems and Computing,2019,vol 841. Springer.

[4] https://zcybersecurity.com/blockchain-in-cybersecurity-use-cases/

[5] Antonakakis, M., Dagon, D., Luo, X., Perdisci, R., Lee, W., & Bellmor, J. *A centralized monitoring infrastructure for improving DNS security*,in International Workshop on Recent Advances in Intrusion Detection,Springer,2010,pp. 18-37.

[6] Ju, Y. W., Song, K. H., Lee, E. J., & Shin, Y. T. *Cache poisoning detection method for improving security of recursive DNS*,in The 9th International Conference on Advanced Communication Technology,IEEE,2010,Vol. 3, pp. 1961-1965.

[7] Park, K., Pai, V. S., Peterson, L. L., & Wang, Z. *CoDNS: Improving DNS Performance and Reliability via Cooperative Lookups,*in *OSDI* ,2004,Vol. 4, pp. 14-14.

[8] Poole, L., & Pai, V. S. *ConfiDNS: Leveraging Scale and History to Improve DNS Security*,in WORLDS,2006.

[9] Sun, H. M., Chang, W. H., Chang, S. Y., & Lin, Y. H. *DepenDNS: Dependable mechanism against DNS cache poisoning*, in International Conference on Cryptology and Network Security, Springer, pp. 174-188.

[10] AlFardan, N. J., & Paterson, K. G. *An analysis of DepenDNS*, in International Conference on Information Security,Springer,2010,pp. 31-38.

[11] Yuan, L., Kant, K., Mohapatra, P., & Chuah, C. N. *DoX: A peer-to-peer antidote for DNS cache poisoning attacks*, in 2006 IEEE International Conference on Communications, IEEE,2006, Vol. 5, pp. 2345-2350.

[12] Perdisci, R., Antonakakis, M., & Lee, W.*Solving the DNS Cache Poisoning Problem Without Changing the Protocol*, Technical report,2008.

[13] Perdisci, R., Antonakakis, M., Luo, X., & Lee, W. *WSEC DNS: Protecting recursive DNS resolvers from poisoning attacks*, in 2009 IEEE/IFIP International Conference on Dependable Systems & Networks, IEEE,2009, pp. 3-12.

[14] Jayasingh, B. B. *DNS Cache Poisoning Attack Analysis and Detection Using Packet Header*, CVR Journal of Science and Technology, 2017,Vol.12, pp.108-112.

*[15]* Lysenko, S., Bobrovnikova, K., Savenko, O., & Shchuka, R. *Technique for Cyberattacks Detection Based on DNS Traffic Analysis*.

[16] Satam, P., Alipour, H., Al-Nashif, Y. B., & Hariri, S. *Anomaly Behavior Analysis of DNS Protocol*, J. Internet Serv. Inf. Secur.,2015, Vol.5, No.4, pp. 85-97.

[17] Sharma, C. *Feed Forward MLP SPAM domain Detection Using Authoritative DNS Records and Email Log*, (Doctoral dissertation, Dublin, National College of Ireland),2020.

[18] Huang, C., Zhang, P., Sun, Y., Zhu, Y., & Liu, Y.*SFDS: A Self-Feedback Detection System for DNS Hijacking Based on Multi-Protocol Cross Validation*, in 2019 26th International Conference on Telecommunications (ICT) ,IEEE, pp. 238-243.

[19] Musashi, Y., Takemori, K., Kubota, S., & Sugitani, K. *Detection of DNS Cache Poisoning Attack in DNS Standard Resolution Traffic*.in CSEC-53,2011.

[20] Ju, Y. W., Song, K. H., Lee, E. J., & Shin, Y. T. *Cache poisoning detection method for improving security of recursive DNS,* in The 9th International Conference on Advanced Communication Technology, IEEE,2007, Vol. 3, pp. 1961-1965.

[21] Maksutov, A. A., Cherepanov, I. A., & Alekseev, M. S. *Detection and prevention of DNS spoofing attacks*,2017 Siberian Symposium on Data Science and Engineering (SSDSE), IEEE,2017, pp. 84-87.

[22] Sahri, N. M., & Okamura, K. *Collaborative Spoofing Detection and Mitigation--SDN Based Looping Authentication for DNS Services*,2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), IEEE,2016, Vol. 2, pp. 565-570.

[23] Wu, H., Dang, X., Zhang, L., & Wang, L.*Kalman filter-based DNS cache poisoning attack detection*, 2015 IEEE International Conference on Automation Science and Engineering (CASE),IEEE,2015,pp. 1594-1600.

[24] Abdelmajid, N., Amin, A., & Farhan, S. A. *Location Based Model for Prevention DNS Spoofing*, Proceedings of the 2020 International Conference on Internet Computing for Science and Engineering,2020, pp. 1-4.

[25] Bassil, R., Hobeica, R., Itani, W., Ghali, C., Kayssi, A., & Chehab, A. *Security analysis and solution for thwarting cache poisoning attacks in the domain name system*, in 2012 19th International Conference on Telecommunications (ICT),IEEE, pp. 1-6.

[26] Cao, J., Ma, M., Wang, X., & Liu, H. *A selective re-query case sensitive encoding scheme against DNS cache poisoning attacks*, Wireless Personal Communications, 2017,Vol.94,No.3,pp.1263-1279.

[27] Hmood, H. S., Li, Z., Abdulwahid, H. K., & Zhang, Y. *Adaptive caching approach to prevent DNS cache poisoning attack*,The Computer Journal, 2015,Vol.58,No.4, pp.973-985.

[28] Fan, L., Wang, Y., Cheng, X., & Li, J. *Prevent DNS cache poisoning using security proxy*, 12th International Conference on Parallel and Distributed Computing, Applications and Technologies, IEEE,2011, pp. 387-393.

[29] Zhao, Y., Hu, N., Zhang, C., & Cheng, X. *DCG: A Client-side Protection Method for DNS Cache*, Journal of Internet Services and Information Security (JISIS),2020, Vol.10,No.2, 103-121.

[30] Mohan, J., Puranik, S., & Chandrasekaran, K. *Reducing DNS cache poisoning attacks*,2015 International Conference on Advanced Computing and Communication Systems, IEEE, pp. 1-6.

[31] Naqash, T., Ubaid, F. B., & Ishfaq, A. *Protecting DNS from cache poisoning attack by using secure proxy*, 2012 International Conference on Emerging Technologies,IEEE,pp. 1-5.

[32] Tzur-David, S., Lashchiver, K., Dolev, D., & Anker, T. *Delay fast packets (dfp): Prevention of DNS cache poisoning.*, in International Conference on Security and Privacy in Communication Systems, Springer, pp.303-318.

[33] Wang, Z., Yu, S., & Rose, S. *An On-Demand Defense Scheme Against DNS Cache Poisoning Attacks*, in International Conference on Security and Privacy in Communication Systems, Springer, 2017, pp. 793-807.

[34] Ma, T., Xu, C., Zhou, Z., Kuang, X., Zhong, L., & Grieco, L. A. *Intelligent-Driven Adapting Defense Against the Client-Side DNS Cache Poisoning in the Cloud*,in GLOBECOM 2020-2020 IEEE Global Communications Conference, IEEE, pp. 1-6.

[35] Chau, S. Y., Chowdhury, O., Gonsalves, V., Ge, H., Yang, W., Fahmy, S., & Li, N. *Adaptive Deterrence of DNS Cache Poisoning,* in International Conference on Security and Privacy in Communication Systems, Springer, 2018, pp. 171-191.

[36] Trostle, J., Van Besien, B., & Pujari, A. *Protecting against DNS cache poisoning attacks*, in 2010 6th IEEE Workshop on Secure Network Protocols, pp. 25-30.

[37] Vixie, P., Gudmundsson, O., Eastlake, D., & Wellington, B. *Secret key transaction authentication for DNS (TSIG)*. RFC 2845,2000.

[38] Eastlake, D. *DNS request and transaction signatures (SIG (0) s)*. RFC 2931, September 2000.

[39] R.Arends, R Austein, M Larson, Daniel Massey, Scott W. Rose. *DNS security introduction and requirement*, IETF,2005.

[40] D.J.Bernstein. *DNSCurve: Usable security for DNS*,2009.

[41] Sun, H. M., Chang, W. H., Chang, S. Y., & Lin, Y. H.*DepenDNS: Dependable mechanism against DNS cache poisoning*,International Conference on Cryptology and Network Security ,Springer,2009,pp. 174-188.

[42] Kumar, N., & Ranga, K. K. *A Framework for Security of DNS using Cryptography*, IJIRST,2015,Vol.2,No.01.

[43] Snigdha, M. S. *A Framework for Security of DNS Using Cryptography*,2020.

[44] Hussain, M. A., Jin, H., Hussien, Z. A., Abduljabbar, Z. A., Abbdal, S. H., & Ibrahim, A. *DNS protection against spoofing and poisoning attacks*,2016 3rd International Conference on Information Science and Control Engineering (ICISCE), pp. 1308-1312.

[45] Jalalzai, M. H., Shahid, W. B., & Iqbal, M. M. W. *DNS security challenges and best practices to deploy secure DNS with digital signatures*, in 2015 12th International

Bhurban Conference on Applied Sciences and Technology (IBCAST), IEEE, pp. 280-285.

[46] Li, X. J., Ma, M., & Arjun, N. *An Encryption Algorithm to Prevent Domain Name System Cache Poisoning Attacks*,29th International Telecommunication Networks and Applications Conference (ITNAC), IEEE,2019, pp. 1-6.

[47] Bai, X., Hu, L., Song, Z., Chen, F., & Zhao, K. *Defense against DNS man-in-the-middle spoofing*, International Conference on Web Information Systems and Mining Springer, 2011, pp. 312-319.

[48] Chen, L., Zhang, Y., Zhao, Q., Geng, G., & Yan, Z. *Detection of DNS DDOS attacks with random forest algorithm on spark*, Procedia computer science, vol.134,2018, pp. 310-315.

[49] Kozlenko, M., & Tkachuk, V. *Deep learning-based detection of DNS spoofing attack*,2019

[50] Moubayed, A., Aqeeli, E., & Shami, A. *Ensemble-based feature selection and classification model for DNS typosquatting detection*,2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 1-6.

[51] Do Xuan, C., Nikolaevich, T. V., Dam, N. Q., Hoang, N. Q., & Long, D. H. *Malicious domain detection based on DNS query using Machine Learning*. International Journal, 2020, Vol.8, No.5.

[52] Jin, Y., Tomoishi, M., & Matsuura, S. *A Detection Method Against DNS Cache Poisoning Attacks Using Machine Learning Techniques: Work in Progress,*2019 IEEE 18th International Symposium on Network Computing and Applications (NCA), IEEE,2019, pp. 1-3.

*[53]* Berger, H., Dvir, A.Z. & Geva, M. *A wrinkle in time: a case study in DNS poisoning ,Int. J. Inf. Secur.* ,2021,Vol. 20,Pp- 313–329.

[54] https://code.google.com/archive/p/omnet-httptools/

[55] Yang, X. S., & He, X., *Firefly algorithm: recent advances and applications*, *International journal of swarm intelligence*,2013,Vol.1, No.1,Pp. 36-50.