Leveraging Deep Learning for Enhancing Cyber Defence Against Pegasus Spyware Attacks

1st Shreya Tiwari
Amity School of Engineering and Technology,
Amity University,
Noida, India
tiwari.shreya.0427@gmail.com or ORCID-0009-0000-8466-5496

2nd Mritunjay Kr. Ranjan
School of Computer Sciences and Engineering,
Sandip university,
Nashik, India.
mritunjaykranjan@gmail.com

3rd Prasad Gadekar
School of Computer Sciences and Engineering,
Sandip university,
Nashik, India.
prasad01198@gmail.com

Abstract—**Advanced cyber threats such as Pegasus Spyware show us why cybersecurity should be unbreachable. This paper attempts to provide an insight into how such cyber defences can be improved using deep learning techniques with help of Pegasus Spyware. In this paper, we provided an investigation on the threat landscape and Pegasus advanced capabilities which leads to proposing a Deep learning-based solution for identifying & neutralizing such sophisticated attacks. Basically, you need neural networks to empower an intrusion detection and prevention system that can withstand threats like Pegasus as well. This effort is critical to advance cybersecurity in support of cyber warfare blending the best practices and processes around digital ecosystems. New digital devices and advancements in technology have led to an increase of cyber threats. Pegasus Spyware is surveillance cyber weapons used for the conduct of targeted attacks and state-sponsored espionage. Pegasus, a military-grade spyware developed by Israeli company NSO Group can be used to monitor remotely on smartphones. It takes advantage of Android and iOS security issues, via which, it is infecting devices. After being installed, Pegasus sideloads onto a device and can collect call records, SMS messages, emails social media activity as well as access the microphone and camera. It can monitor location, intercept encrypted messages and exfiltrate any data. Malware is tough to find due obfuscation, encryption and zero-day exploits. Pegasus used to be marketed just to law enforcement, but now it has been employed against journalists and opposition activists (as well as political dissidents), leaving questions over its morality. Pegasus spyware illustrates the advanced functionality, potential for abuse, and continuous evolution of cyber weapons, highlighting the critical need for measures to protect internet and personal data.**

Keywords— Deep Learning, Cyber Defence, Pegasus Spyware Attacks, cyber weapon, Surveillance, Sophisticated device

I. Introduction

The cybersecurity world is changing constantly and advanced cyber weapons like Pegasus Spyware, are a significant threat. In this study, deep learning is investigated as a technology that can bolster defences against advanced threats [1]. We intend to develop efficient intrusion detection and prevention systems by analysing Pegasus with respect to the threat landscape. To develop defence methods, we employ neural networks to secure information systems from extremely tough adversaries performing complex attacks including Pegasus Spyware. Internet is now as common to use in daily routine. Therefore, it will be easier for criminals to invade privacy lanes, obtain information without permission and misuse it. With Pegasus Spyware on the rise and so many other sophisticated cyber threats, new forms of cybersecurity defence are a must. Pegasus is an extremely covert surveillance program made by NSO Group that turns mobile devices into spy cameras, which can subvert smartphones and steal sensitive information as well as monitoring people without the targets knowing what was happening [2],[3]. Advanced threats are difficult to detect and prevent with traditional cybersecurity methods because of their complexity, as well as evasiveness. Novel cyber defences against Pegasus and other malicious entities are so required. Deep learning leverages neural networks to identify anomalies, classify malware and enhance intrusion detection & prevention systems leading towards benefits for cyber security [4]. This paper presents the employment of deep learning to strengthen cyber defence mechanisms and mitigate Pegasus Spyware based attacks to dim down risks such powerful cyber weapon imprints [5]. Cyber threats such as the Pegasus Spyware continue to grow and advance, this calls for new and reliable cybersecurity defences. However, deep learning could help bolster cyber defences against sophisticated attacks as it enables more accurate and faster threat detection and remediation [6]. The project looks at leveraging deep learning against Pegasus and other sophisticated cyber weapons to harden the digital systems & protect user privacy in a highly vulnerable digital ecosystem.

Objective:

- To analyse the capabilities, operation modes, and potential impacts of Pegasus Spyware on targeted systems.
- To explore deep learning methodologies, such as neural networks, for developing effective intrusion detection and prevention mechanisms against Pegasus Spyware.
- To evaluate and compare the performance of deep learning-based defence mechanisms with traditional cybersecurity approaches in detecting and preventing Pegasus Spyware infections

II. Related Study

Many ransomwares, viruses, spywares, and other malicious software attack people, spy on their online behaviour, and steal their data and personal information. Adware, trojan horses, worms, botnets, and ransomware also exist to pose various user dangers. The most frequent computer malwares are

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

Ransomware, Computer viruses, Spywares, Worms, Trojan horse, Adware.

Table 1. Comparison: Category, Methods, Goal, Example & Gap.

Category	Method	Primary Goal	Example	Research Gap
Ransomware [7]	User action or RaaS	Extortion through data/device lock	WannaCry, Cryptolocker	Create ransomware detecting systems before encryption.
Computer Viruses [8]	User actions	Data destruction, resource usage, propagation	ILOVEYOU, Mydoom	Improve antivirus software to handle new viral signatures.
Spyware [9]	Installed without user consent	Stealing information, tracking activities	Pegasus, Keyloggers	Developing spyware detection and privacy protection tools.
Worms [10]	Network (e.g., LAN)	Spreading itself, stealing data, installing backdoors	SQL Slammer, Conficker	Increase network security to prevent worms and vulnerabilities.
Adware [11]	Installed covertly with other software	Displaying ads, generating revenue through clicks	Fireball, DollarRevenue	Improve adware detection and removal without affecting legitimate software.
Spyware (Pegasus) [12]	Installed without user consent	Stealing information, surveillance	Pegasus	Improved Pegasus identification and real-time monitoring.

Malware that locks data or devices until a ransom is paid. Can be spread through RaaS agreements where developers share their code with hackers [13].

A programme that can modify another programme to include a copy of itself, propagating through infected files. Requires user action to spread [14].

Malware installed without user knowledge, stealing sensitive information and tracking internet usage. Often peeks into data and activities [15].

Self-replicating malware that spreads without human action, often exploiting network vulnerabilities [16].

Malware disguised as legitimate software, often delivered through social engineering [17].

Malware that displays intrusive ads, sometimes tracking user activity for targeted advertising [18].

Highly sophisticated spyware tracking user activities and stealing data from target devices [19].

III. PURPOSED METHODOLOGY

Model Training and Optimization for Deep Learning Use metrics accuracy, precision recall and F1 score to evaluate models. Take top model and add those to a anomaly detector + monitoring (real time analysis) Find out more on … Adaptive with Feedback Loop and Privacy-Focused The new tactic will help in defending against Pegasus spyware[20],[21].

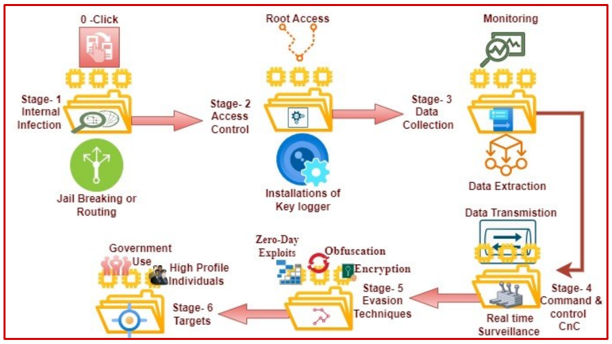


Fig. 1. Methodology to move forward.

A. Pseudocode

As per (Fig.1.) the pseudocode is:

```
graph TD;
A [Zero-Click Exploit] →B [Jailbreaking/Rooting]
B →C [Root Access]
C →D [Installation of Keylogger]
C → E [Monitoring Camera, Microphone, and Apps]
D → F [Data Collection: Calls, Texts, Location, Pictures,
        Passwords]
F → G [Data Transmission to CnC Server]
G → H [Real-Time Surveillance]
H → I [Targets: Journalists, Dissidents, Politicians,
        Government Officials, Human Rights Advocates]
I → J [Government Use for Security and Law Enforcement]
C →K [Evasion Techniques]
K→ L [Obfuscation]
K → M [Encryption]
K → N [Zero-Day Exploits]
```

B. Initial Infection

Pegasus infects the target device without the user's knowledge by using zero-click exploits. automatically roots or jailbreaks the device to have complete access.

C. Access and Control

Circumvents security precautions by gaining root access to the device. Installs a keylogger to record passwords and login information.

D. Command and Control (CnC)

Returns gathered info to the CnC server. gives instant access to the victim's information and actions.

E. Evasion Techniques

Obfuscates malware code to hide. Data is encrypted to hide from security software. constantly exploits weaknesses to escape detection.

F. *Targets*

Focuses on human rights activists, legislators, journalists, dissidents, and government officials. Governments say they use it for law enforcement and security.

IV. Deep learning Approach

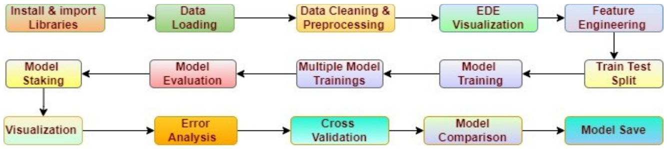


Fig. 2. Fig. Layout of Deep learning Approach

A. *Algorithm*

As per (Fig.2.) DL algorithm is:

S1. Installation of necessary python libraries.

S2.Load the dataset containing features and labels indicating Pegasus attack stages and normal behaviour.

S3. Data cleaning & Preprocessing.

S3.1.Feture Engineering.

S3.1.1. Handling Class Imbalance.

S3.1.2. Feature Selection.

S3.1.3. Splitting Data into Features (x) and Target(y).

S4. Train Test Split.

S5. Model Training.

S.5.1. RandomForestClassifier.

S5.2. GradientBoostingClassifier

S5.3.XGBClassifier

S5.4. NeuralNetwork

S6. Model evolution.

S6.1. Evaluation Metrics

S6.2. Confusion Matrix

S6.3. ROC Curve

S6.4. Precision-Recall Curve

S7.Staking Model.

S8. Visualization.

S8.1. Distribution of Predictions

S9.Error Analysis

S10. Comparison

S11. Model Save

B. *Formalize Mathematically*

The three main elements of the model, the feature extraction procedure, the training procedure, and the assessment metrics in order to mathematically codify the machine learning technique for identifying Pegasus spyware.

1) *Feature Extraction*

A data set X with n sample size and m features, where a feature xi is extracted for each sample i.

$$x_i = [x_{i1,} x_{i2,} \dots\dots\dots\dots.., x_{im,}] \qquad (i)$$

For each feature vector, Specific features related to the Pegasus lifecycle, extracted say f_i.

$$f_i = ExtractFeatures\,(x_i) \qquad (ii)$$

2) *Model Training*

Let F be the matrix of extracted features for all samples and y be the vector of all levels.

$$F = [\,f_1, f_2, \dots\dots\dots, f_n\ ]^T \qquad (iii)$$

$$Y = [\,y_1, y_2, \dots\dots\dots, y_n\ ]^T \qquad (iv)$$

RandomForestClassifier model will be trained on F and y, the model parameters are denoted as θ [22].

$$\theta = TrainModel(F, y) \qquad (v)$$

GradientBoostingClassifier model Will be trained on M titration is[23].

$$F_m(x) = F_0(x) + \sum_{m-1}^{m} \propto_m h_m(\mathrm{x}) \qquad \text{(vi)}$$

For binarily classification the prediction will be

$$P(y = 1\,|x) = \frac{1}{1 + e^{-F_m(x)}} \qquad (vii)$$

Where,

$F_{0(}$x) is the initial model.

α_m is the m- th weak leaner.

h_m(x) is the m-th weak leaner.

XGBClassifier model will be trained after M titration[24].

For binary classification, the prediction probability:

$$F_M(x) = F_0(x) + \sum_{m-1}^{M} \propto_m h_m(\mathrm{x}) \qquad \text{(viii)}$$

$$P(y = 1\,|x) = \frac{1}{1 + e^{-F_m(x)}} \qquad (ix)$$

Where,

$F_{0(}$x) is the initial prediction, often set to zero (0) or the mean log-odds of the training labels.

α_m is learning rate.

h_m(x) is the m-th weak leaner.

NeuralNetwork, the final output with L layer is[25].

$$Z^L = W^L a^{L-1} + b^L \qquad (x)$$

Where,

a^0 = x(input features)

W^l and b^l are the weights and biases for layer l.

$z^l = W^l a^{l-1} + b^l$ is the linear transformation at layer l

$a^l = \sigma(z^l)$ is the activation function.

3) Prediction

For a new sample x_j, features is extracted and predictions take place with respect to label trained model.

$$f_i = ExtractFeatures\ (x_j) \qquad (xi)$$

$$\hat{y}_j = Predict(f_j; \theta) \qquad (xii)$$

1) Prediction

The model's performance is evaluated using metrics such as accuracy, precision, recall, and F1- Score.

Let ŷ be the vector of predicted labels for the test set y_{test} be the true levels.

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n} 1(\hat{y}_i = y_{test,i}) \qquad (xiii)$$

$$Precision = \frac{TP}{TP+FP} \qquad (xiv)$$

$$Recall = \frac{TP}{TP+FN} \qquad (xv)$$

$$F1 - Score \ = \frac{2 * Precision * Recall}{Precision * Recall} \qquad (xvi)$$

Where,

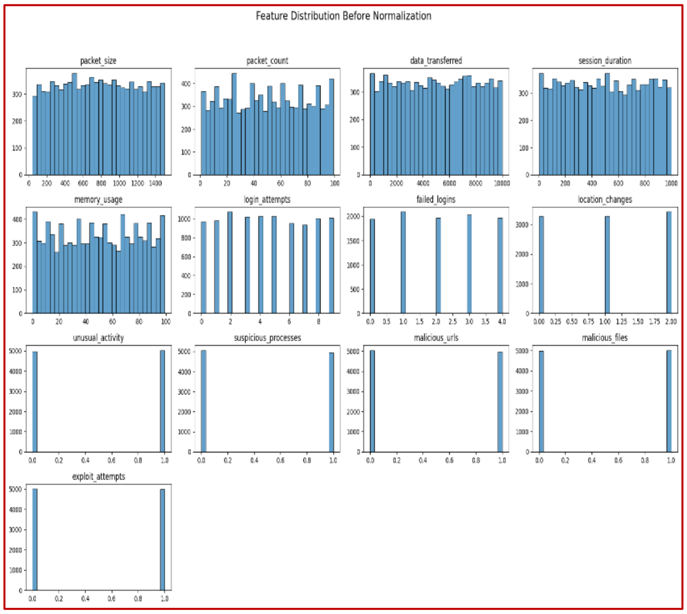
TP = True Positives

FP = False Positives

FN = False Negative

Based on the given workflow, this mathematical formulation describes the essential procedures and measurements needed to create a machine learning system that can identify Pegasus spyware. To match a dataset and detection needs, feature extraction and model parameters can be changed as necessary.

V. Result, Visualization and Discussion



Fig,3, Feature Distribution before Normalization

The Noval data set (Cyber Attack Dataset) Dispersion of the features in image displaying how they provide the patterns and variations which are useful to identify Pegasus spyware. These features can be used to build a deep learning model aid in defence against cyber threats and provide better behavioural detection of malware activities. As a result, the likelihood of Pegasus malware attacks is reduced.

Before normalization, the (Fig.3) shows a figure that represents how feature distributions are in our dataset. we must implement these features to build a strong machine learning model and able it can detect Pegasus malware. These include network activity and system usage, as well as security events. It is beneficial to know these distributions and an array of other distributions available as this allows for the recognition patterns in irregular installations that could indicate spyware on a device.

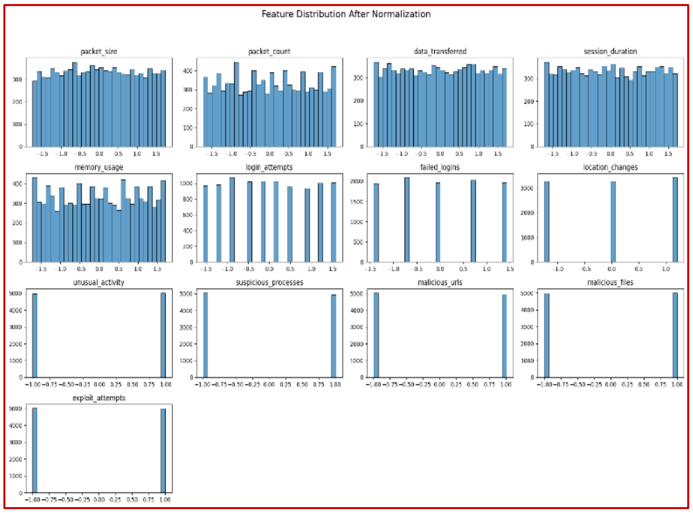


Fig.4. Feature Distribution after Normalization

The (Fig 4) well illustrates the accomplished distribution of features after Normalization and points out how data was prepared for training model trained using Deep Learning. Since all features are standardized, it allows a model trained on this data set to learn more effectively and hence makes more accurate predictions in increasing cyber defence against spywares such as Pegasus.

We can investigate (Fig.5) a feature importance chart to identify top features which the Random Forest classifier believe are important for identifying attacks of Pegasus spyware. Network related characteristics such as source and destination ports, exchanged data, system calls are some of the most important in identifying suspicious activities associated with spyware. The use of these features and the benefits accruing for finding malware is prominently proven. It may help to refine the model to focus on critical indications, and ideally concert with enhanced cyber defence practices.

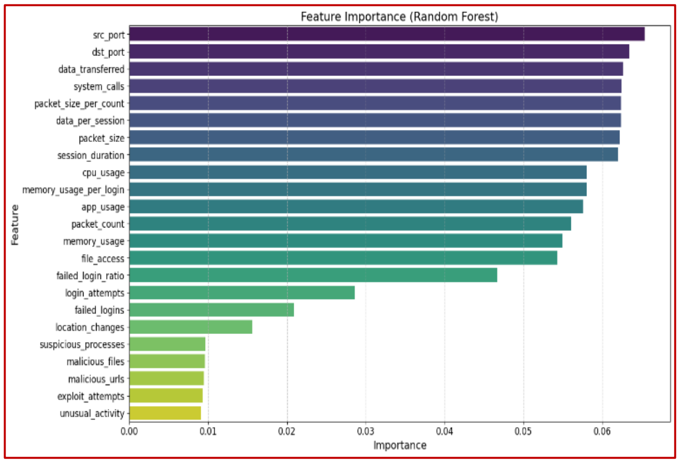


Fig.5. Feature Importance (Random Forest)

The (Fig.6) is a learning curve representing the training and validation loss over epochs when you train a machine learning model. The X-axis represents the epochs, and the Y-axis refers to loss values that indicate how accurate our forecast was. Two of the losses are high at first (indicating bad initial predictions). Adequate number of epochs also shows a rapid decrease in losses to almost negligible values, thus hinting towards model drastically improving when new data is being passed. After about twenty epochs, the losses stabilize at low values showing that model has made converged and take good lessons. Minimal and stable losses across both the training data used to feed into the network as well as on the validation datasets during that time will indicate a strong fit without overfitting. Overall, the learning curve indicates that training has worked and the model can detect Pegasus infections.

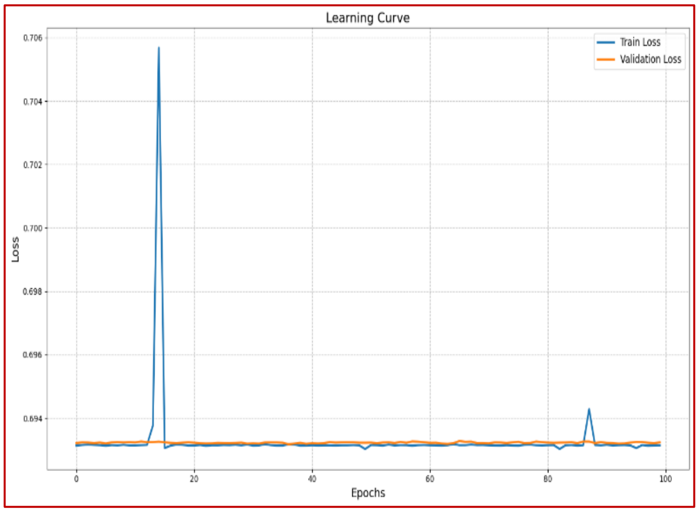


Fig.6. Learning Curve

Accuracy and validation accuracy over 100 epochs of deep learning model for Pegasus spyware attacks prevention in cyber security. At first, we see that both losses are the same which implies learning. Although there is a spike in training loss around epoch 20, it appears that this might just be an abnormal transient (as whole model distribution does not change and goes back to its normal state quickly). This represents a strong generalization and few overfittings since it would mean that the training loss always aligns with validation gain throughout the study. Overall, the model performs with trustable results hinting its help in robust detection and prevention of Pegasus spyware, which on way strengthens cyber defence.

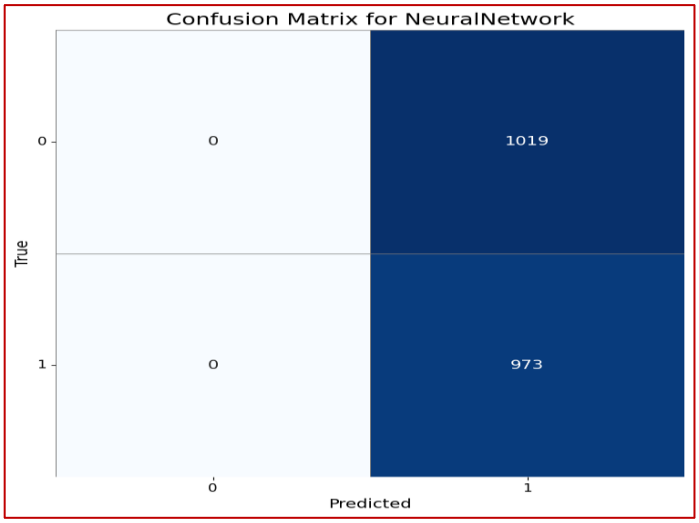


Fig. 7. Confusion Matrix for Neural Network

Neural Network Classifier Confusion Matrix (Fig7).: (5285, 1019) - Negatives; Negative predictions had almost no cost attached to them. This means that the model is very heavily biased toward predicting class 1 (as we can see by having no true negatives) Although the model did detect all true positives (973, yielding a recall of 1 for class 1), it could not predict any true negatives, making its predictions of zero same as number_of_false_negatives. This means that the precision will be weak, as due to errors in classifying high number of positive cases. It requires further tweaking of the model and perhaps more balanced data for training to be able to accurately identify both groups. The model is performing not consistently overall and need some tweaking.

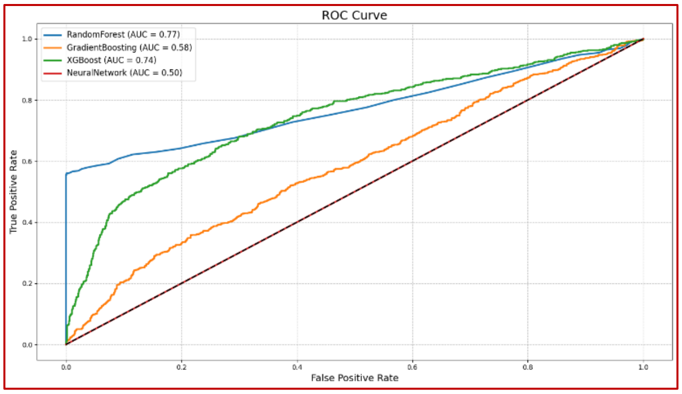


Fig. 8. ROC Curve

This ROC curve (receiver operating characteristic) (Fig.8.) evaluates a variety of classifiers to improve the cyber defensive mechanism against Pegasus spyware attacks. And this assessment is in expansion to the target of utilizing profound learning. The X-axis on the graph shows false positive rate, while Y gives a proportion of actual positives. The best performing classifier in this dataset is Random Forest which has an AUC 0.77.offsetWidth This means it has a high precision or true positive and low false positives. There is also a solid performance from XGBoost model that achieves Area under the curve (AUC)=0.74, which provides good detection capabilities toward malware as well. Gradient Boosting is somewhat better than KNN but significantly worse compared to Random Forest and XGBoost with an area under the curve (AUC) of 0.58 The Neural Network, on the other hand lands in a whopping 0.50 AUC range - this means their predictions are basically pure chance and do you know what is regarded as poor? In summary, the ROC curve analysis in this study revealed that although deep learning techniques such as Random Forest and XGBoost can greatly strengthen cyber security against Pegasus malware attacks, more effective development of Neural Network is required. Hence, the need to build robust models such as Random Forest and XGBoost for maximum Brooklyn cyber protection. This definition of the target model is ideal to find a compromise between recognizing as many true positives as possible and minimizing false positive rate.

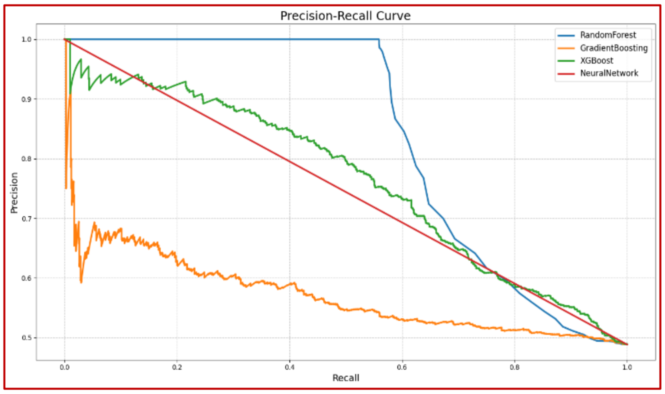


Fig.9. Precision- Recall Curve

The presented Precision-Recall curve (Fig 9) is used to evaluate the performance of multiple classifiers (Random Forest, Gradient Boosting, XGBoost and Neural Network), in a use case where deep learning can be employed for enhancing cyber defence against Pegasus spyware attacks. The Y-axis is precision, and along the X-axis you have recall. Random Forest is the best performing model of all experimental thresholds in terms when detecting spyware attacks, being accuracy-wise largest across recall levels. This makes it the optimal classifier out of all classifiers. Gradient Boosting demonstrates a fair amount of precision, which drops off as recall gets better indicating good performance. XGBoost starts in a high precision state, but as recall goes up (more and more positives are discovered), the precision declines very fast - this is an important aspect showing that it might have issues scaling to many positives. As we considered the performance of the Neural Network is bad, and it reflects on low precision for all recall values. This research shows that Random Forest is best from the label data to enhance cyber defence against Pegasus malware and can be used as a future attack model. This provides a good balance of high precision and recall, so it is the best classifier. Principal components are valuable for logistic regression and decision-tree-based approaches, but the neural network needs more work to reach usable accuracies; gradient boosting and XGBoost perform respectably. Adoption of robust models, like Random Forest significantly enhances the accuracy and confidence in identifying Pegasus spyware, providing an added layer to cyber defence measures.

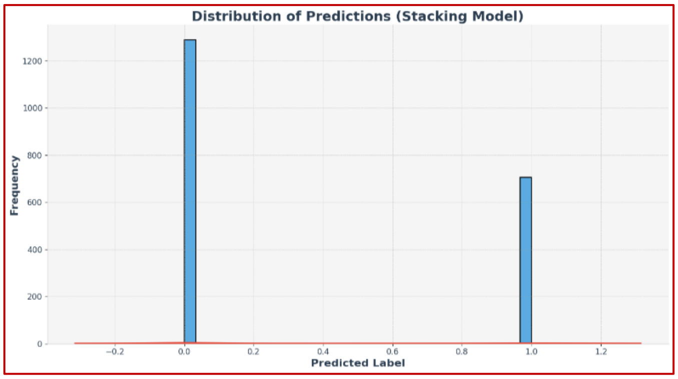


Fig.10. Staking Model

The given bar chart (Fig.10.) visualizes the predictions distribution of a strong stacking model for which intends to empower cybersecurity against Pegasus spyware using deep learning utilities. This model is very good at detecting true spyware attacks, since it has been able to achieve an accuracy of 74.30%, as indicated by a precision value of 0.82 that classifies the most occurrences accurately. The recall of 0.61, this is a sign that some real positives are not identified as such. The F1 score is 0.70 which means that both precision and recall were kept at a good level (rather than one being negligibly small). The stacking model is an effective mechanism for improving spyware detection in general. It offers highly secured defence mechanisms; however, it still needs to completely mature with a good detection model of actual true positives.

We have the bar chart (Fig.11) which provides performance comparison of Random Forest, Gradient Boosting,, XGBoost, Broad Network and Stacking models in accuracy precision recall F1-score They show the power of these models with taking deep learning solutions in cyber defence for upgrading against Pegasus spyware attacks. The Random Forest still has a respectable precision (0.77) and broader F1-score of 0.70, however concluding that the model is not fragile to be applicable only in cases where spyware attacks need detection but can also identify such cases properly. Conclusion: Gradient Boosting

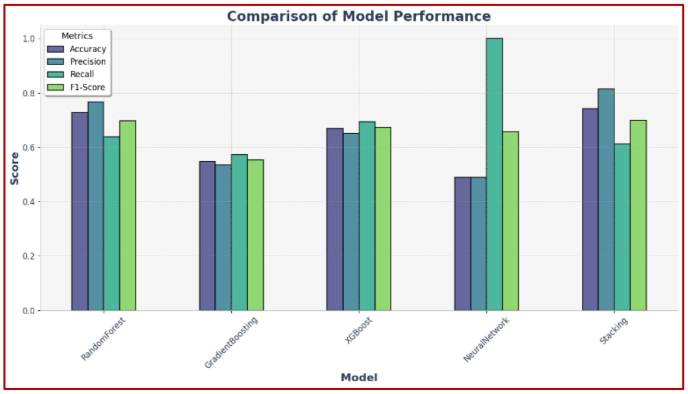


Fig.11. Comparison of Model Performance

Slope with an accuracy of 0.55 and F1-Score also provided a release mean number of competences, so it helps in smooth generalization like there is the Gap between train and test accuracies but higher than Naive base line algorithm model age based on just domain knowledge. XGBoost is most relevant with balanced metrics scored an accuracy of 0.67, F1-score of 0.67 The Neural Network shows an extremely high recall (1.00), but its precision/accuracy is very low with 0.49 for each of them, meaning it creates a lot of false positives. The Stacking model performs the best in terms of precision (0.82), accuracy (0.74) and balanced F1-score which is improved by efficiently merging various models to enhance spyware detection efficiency along with diversity, more specifically, experimental results show that Stacking ensemble model, Random Forest and XGBoost are three of the best models for Pegasus malware prediction. These techniques result in the enhanced detection of any kind and thereby improving cyber defensive methodologies. Although the Neural Network must be finely tuned - because it has a very high level of false positives, results achieved with Gradient Boosting are only ok.

VI. Conclusion

The analysis and visualizations that have been provided prove how deep learning can make interception of Pegasus spyware more efficient with the help (but not completely exclusive) to those number of models. Among them, the stacking model turns out to be useful with an accuracy of 74.30%, precision:0.82; recall:0.61; f1-score: 0.70 This clearly indicates that it is working in a perfect trade-off between Precision and Recall detecting Spyware very well. The high precision means it can correctly identify a majority of occurrences and the moderate recall suggests that some true positives are missed. The stacking model, Random Forest and XGBoost are the top 3 best performing models among all three datasets for spyware detection as shown in Table. Random Forest showed a promising performance, with its score between 0.70-73, and XGBoost found the best balance in metrics accuracy (with F1-score), both around ~67 %. The Neural Network has a high recall (1.00), but poor precision and accuracy (0.49) results in large amounts of false positives. On the other hand, Gradient Boosting displays average effectiveness with an accuracy and F1-score of 0.55 respectively which becomes quite acceptable but less optimal performance. This stacking model has also made good usage of different models to bootstrap better detection efficiency and diversity, thereby providing the best precision rate with respect to other balanced metrics. This underscores a powerful capability related to increasing the robustness of spyware detection and improving cyber-security defences. Its many advantages seem to be making it the default method of choice, but a good degree tuning is still needed in case like Neural Network for reducing number false positives. Cyber defence against Pegasus malware is greatly improved with an additional layer of deep learning, especially through models such as stacking and Random Forest or XGBoost. They enhance the cybersecurity policy as these models enables dependable detection. Further developments and refinements of these models will continue to improve however the development phase is already providing significant improvement in defending against complex cyber threats.

VII. Future Scope

The scope for future exploitation of deep learning for the purpose of strengthening cyber defence against Pegasus spyware Broadening detection across devices and operating systems, enabling more explainability as well as supportive defence methods can go a long way in making cybersecurity better. To keep models relevant and reliable, we need to implement continuous learning systems that concentrate on resource-efficient methods. For this reason, addressing legal and ethical considerations will be essential for innovative technologies to take off while maintaining the security of cybersecurity defences.

References

[1] K. Thakur, M. Qiu, K. Gai, and M. L. Ali, “An Investigation on Cyber Security Threats and Security Models,” IEEE Xplore, 2019. https://ieeexplore.ieee.org/abstract/document/7371499

[2] “NSO Group in trouble again over spyware,” Network Security, vol. 2021, no. 8, pp. 2–3, Aug. 2021, doi: https://doi.org/10.1016/s1353-4858(21)00082-9.

[3] M. Sparkes, “UK targeted by spyware,” New Scientist, vol. 254, no. 3383, p. 7, Apr. 2022, doi: https://doi.org/10.1016/s0262-4079(22)00680-7.

[4] A. Liu, Y. Ren, Z.-H. Pang, and B. Niu, “Fixed-Time Secure State Estimation for Cyber-Physical Systems With Multi-Channel Transmission Under DoS Attacks,” IEEE Transactions on Circuits & Systems II Express Briefs, pp. 1–1, Jan. 2024, doi: https://doi.org/10.1109/tcsii.2024.3370400.

[5] Z. Rehman, I. Gondal, M. Ge, H. Dong, M. A. Gregory, and Z. Tari, “Proactive Defense Mechanism: Enhancing IoT Security through Diversity-based Moving Target Defense and Cyber Deception,” Computers & Security, pp. 103685–103685, Jan. 2024, doi: https://doi.org/10.1016/j.cose.2023.103685.

[6] M. Al-Hawawreh and M. S. Hossain, “A privacy-aware framework for detecting cyber attacks on internet of medical things systems using data fusion and quantum deep learning,” Information Fusion, vol. 99, p. 101889, Nov. 2023, doi: https://doi.org/10.1016/j.inffus.2023.101889.

[7] Sekione Reward Jeremiah, H. Chen, Stefanos Gritzalis, and Jong Hyuk Park, “Leveraging application permissions and network traffic attributes for Android ransomware detection,” Journal of Network and Computer Applications, pp. 103950–103950, Jun. 2024, doi: https://doi.org/10.1016/j.jnca.2024.103950.

[8] J. Singh, D. Kumar, Z. Hammouch, and A. Atangana, “A fractional epidemiological model for computer viruses pertaining to a new fractional derivative,” Applied Mathematics and Computation, vol. 316, pp. 504–515, Jan. 2018, doi: https://doi.org/10.1016/j.amc.2017.08.048.

[9] K. M. E. N. Mallikarajunan, S. R. Preethi, S. Selvalakshmi, and N. Nithish, “Detection of Spyware in Software Using Virtual Environment,” 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Apr. 2019, doi: https://doi.org/10.1109/icoei.2019.8862547.

[10] Q. Gao and J. Zhuang, “Stability analysis and control strategies for worm attack in mobile networks via a VEIQS propagation model,” Applied Mathematics and Computation, vol. 368, pp. 124584–124584, Mar. 2020, doi: https://doi.org/10.1016/j.amc.2019.124584.

[11] S. Seraj, M. Pavlidis, Marcello Trovati, and Nikolaos Polatidis, “MadDroid: malicious adware detection in Android using deep learning,” Journal of cyber security technology, pp. 1–28, Aug. 2023, doi: https://doi.org/10.1080/23742917.2023.2247197.

[12] S. D. Kaster and P. C. Ensign, “Privatized espionage: NSO Group Technologies and its Pegasus spyware,” Thunderbird International Business Review, vol. 65, no. 3, Dec. 2022, doi: https://doi.org/10.1002/tie.22321.

[13] O. Or-Meir, N. Nissim, Y. Elovici, and L. Rokach, “Dynamic Malware Analysis in the Modern Era—A State of the Art Survey,” ACM

Computing Surveys, vol. 52, no. 5, pp. 1–48, Sep. 2019, doi: https://doi.org/10.1145/3329786.

[14] R. HRISTEV, M. VESELINOVA, and K. KOLEV, "Ransomware Target: Linux. Recover Linux Data Arrays after Ransomware Attack.," The Eurasia Proceedings of Science Technology Engineering and Mathematics, vol. 19, pp. 78–86, Dec. 2022, doi: https://doi.org/10.55549/epstem.1219172.

[15] A. K. Jain, S. R. Sahoo, and J. Kaubiyal, "Online social networks security and privacy: comprehensive review and analysis," Complex & Intelligent Systems, vol. 7, no. 5, Jun. 2021, doi: https://doi.org/10.1007/s40747-021-00409-7.

[16] S. Valizadeh and M. van Dijk, "MalPro," Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop - CCSW'19, 2019, doi: https://doi.org/10.1145/3338466.3358920.

[17] S. Abraham and I. Chengalur-Smith, "An overview of social engineering malware: Trends, tactics, and implications," Technology in Society, vol. 32, no. 3, pp. 183–196, Aug. 2010, doi: https://doi.org/10.1016/j.techsoc.2010.07.001.

[18] K. Subramani, X. Yuan, O. Setayeshfar, P. Vadrevu, K. H. Lee, and R. Perdisci, "When Push Comes to Ads," Proceedings of the ACM Internet Measurement Conference, Oct. 2020, doi: https://doi.org/10.1145/3419394.3423631.

[19] E. Liu et al., "No Privacy Among Spies: Assessing the Functionality and Insecurity of Consumer Android Spyware Apps," Proceedings on Privacy Enhancing Technologies, vol. 2023, no. 1, pp. 207–224, Jan. 2023, doi: https://doi.org/10.56553/popets-2023-0013.

[20] M. K. Qabalin, M. Naser, and M. Alkasassbeh, "Android Spyware Detection Using Machine Learning: A Novel Dataset," Sensors, vol. 22, no. 15, p. 5765, Aug. 2022, doi: https://doi.org/10.3390/s22155765.

[21] M. S. Akhtar and T. Feng, "Malware Analysis and Detection Using Machine Learning Algorithms," Symmetry, vol. 14, no. 11, p. 2304, Nov. 2022, doi: https://doi.org/10.3390/sym14112304.

[22] Manish Choubisa, R. Doshi, N. Khatri, and Kamal Kant Hiran, "A Simple and Robust Approach of Random Forest for Intrusion Detection System in Cyber Security," May 2022, doi: https://doi.org/10.1109/icibt52874.2022.9807766.

[23] K. Omari, "Phishing Detection using Gradient Boosting Classifier," Procedia Computer Science, vol. 230, pp. 120–127, Jan. 2023, doi: https://doi.org/10.1016/j.procs.2023.12.067.

[24] K. M. K. Raghunath, V. V. Kumar, M. Venkatesan, K. K. Singh, T. R. Mahesh, and A. Singh, "XGBoost Regression Classifier (XRC) Model for Cyber Attack Detection and Classification Using Inception V4," Journal of Web Engineering, Apr. 2022, doi: https://doi.org/10.13052/jwe1540-9589.21413.

[25] X. A. Larriva-Novo, M. Vega-Barbas, V. A. Villagrá, and M. S. Rodrigo, "Evaluation of Cybersecurity Data Set Characteristics for Their Applicability to Neural Networks Algorithms Detecting Cybersecurity Anomalies," IEEE Access, vol. 8, pp. 9005–9014, 2020, doi: https://doi.org/10.1109/ACCESS.2019.2963407.