Automatic Method To Predict And Classify Cyber HackingBreaches Using Deep Learning

Mr. K. Ramesh[1*], Dr.T.V.Prasad[2],V. Krishna kanth[3], V. Sarveswar[4],
SD. Abrar Ali[5], M. Vikas Reddy[6]

[1,2,3,4,5,6]Department Computer Science and Engineering, Godavari Institute
Of Engineering and Technology(A), Rajamahendravaram-533296, A.P.
India.

[*1]kothpalliramesh@gmail.com,[3]krishnak.vemuri03@gmail.com,
[4]sarveswarareddyvemulapati@gmail.com,
[5]syedabrarali209@gmail.com,[6]reddyvikas764@gmail.com

Abstract:

Analyzing cyber incident datasets is crucial for enhancing our comprehension of the evolving cyber threat landscape. In present generation many cyber hacking breaches taking place. This Application, Examine diverse cyber-attacks and breaches, analyze the methods employed in these incidents, and explore alternative strategies for prevention. This Application show that rather than by distributing these attacks as because they exhibit autocorrelations, traditional methods for breach prediction and classification, such as rule-based systems and signature-based approaches, have notable limitations. In this application, both prediction and classification of cyber-attacks is done to avoid indeed getting worse in terms of their frequency. This Application will make use of algorithms such as Convolution Neural Network (CNN) and Recurrent Neural Network (RNN) for analyzing our results. In this Application we will be analyzing two types of attacks U2R and R2L attacks.

Keywords: Convolution Neural Network (CNN), Recurrent Neural Network (RNN), Breaches, User to Root(U2R), Remote toLocal(R2L).

1 Introduction

Cyber hacking breaches have become an increase concerning issue in today's digital world. Cyber Attacks or breaches involve unauthorized access to computer systems or networks, often with malicious intent, leading to the compromise of sensitive data, disruption of operations and significant financial and reputational damage. Cybercriminals employ a variety of sophisticated techniques, such as malware, phishing attacks, and exploitation of software vulnerabilities, to breach the security measures put in place by individuals, organizations, and even governments. Cyber-attacks must be predicted and classified to reduce the attacks indeed getting worst in terms of the frequency. Predicting and classifying cyber-attacks using deep learning is a cutting-edge approach in the field of cyber security. To predict and classify cyber-attacks, one can utilize deep learning techniques like Recurrent Neural Networks (RNNs) or Long Short-Term Memory networks. (LSTMs) to analyze sequences of network traffic or system log data. This application prefers RNN rather than CNN for making accurate results.

Deep learning models really shine when it comes to sorting out different types of cyber threats. They're like experts at spotting a sneaky malware attack, putting up a red flag for a potential phishing email, or picking up on suspicious patterns that suggest unauthorized access. These algorithms are super accurate at categorizing incidents, helping us prioritize our responses and deal with risks in a really effective way.

In a nutshell, tapping into the power of deep learning to foresee and classify cyber hacking breaches represents a game-changing shift in how we approach cyber security. Organizations, armed with neural networks and advanced analytics, can beef up their defenses and stay one step ahead in the constantly evolving world of cyber threats. As technology continues to advance, weaving deep learning into our cyber security strategies becomes increasingly crucial – it's like giving our digital guardians a supercharged

K. R. Madhavi et al. (eds.), *Proceedings of the International Conference on Computational Innovations and Emerging Trends (ICCIET 2024)*, Advances in Computer Science Research 112,
https://doi.org/10.2991/978-94-6463-471-6_114

boost, a key player in keeping our sensitive information safe and making sure our digital systems can bounce back from anything.

2 Literature Survey

Shan Sutharan Et al. in 2012 has proposed ″An Iterative ellipsoid-based anomaly detection technique for Intrusion Detection system", implemented based on ellipsoid based technique to detect anomalies, KDD'99 and NSL-KDD datasets are used to detect the Anomalies [1].

Rafath Samrin Et al. in 2017 has proposed″Review on network Anomaly based system", investigation of different techniques and intrusion classification on KDD Cup 99 dataset [2].

Pengtian Chen Et al. in 2022 has proposed″Research on Intrusion Detection Model based on Bagged Tree", implemented based on Bagged Tree and used dataset UNSW_NB15 is used to verify the model [3].

Chaofan Lu Et al. in 2022 has proposed"Research on Technical application of Artificial intelligence in Network Intrusion Detection System", implemented based Artificial Intelligence techniques like neural networks, neural algorithms for automatic detection of Intrusions in the network or Network Intrusion [4].

Ajmeera Kiran Et al. in 2023 has proposed"Intrusion Detection System Using Machine Learning", implemented using machine learning algorithms as a possible paradigm for development of network Intrusion Detection System [5].

3 Existing System

3.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are like the Sherlock Holmes of cybersecurity—they're masters at spotting patterns in complex data. These deep learning models are perfect for sleuthing through network traffic, analyzing malware, and sniffing out intrusions. They're like cyber-detectives, trained to zoom in on the unique traits of network packets, files, or even snippets of code to uncover anything fishy. By sifting through raw data, they can tell the difference between normal and suspicious activities, even if they've never seen the threat before. Plus, they're pros at handling mountains of data, which is crucial in cybersecurity where you're swimming in information. And the best part? They're always learning and adapting, keeping up with the bad guys' latest tricks.

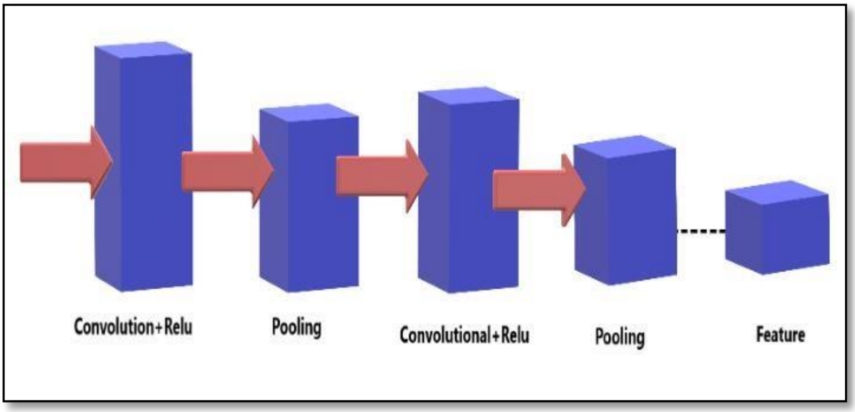


Figure 1: CNN Model

4 Proposed System

4.1 Recurrent Neural Networks

A type of artificial neural network, the Recurrent Neural Network (RNN), displays temporal dynamic behavior through node connections in a sequential graph. The Application is User friendly, can predict results with less time consumption and also provides High Output Efficiency.

It also has a web Interface that contains Input fields related to Network Anomaly detection dataset. Based on the input values like password length, tcp protocol length, flag etc. These input values are sent to the RNN model for prediction of results.

By using LSTM (Long Short-Term memory) we can analyze the patterns effectively and can predict the cyber-attacks accurately. So if there are any new values entered by the user, RNN model itself will train accordingly to find new patterns for the existing one.

An RNN model is modeled to remember each information throughout the time which is very help full in any time series predictor.

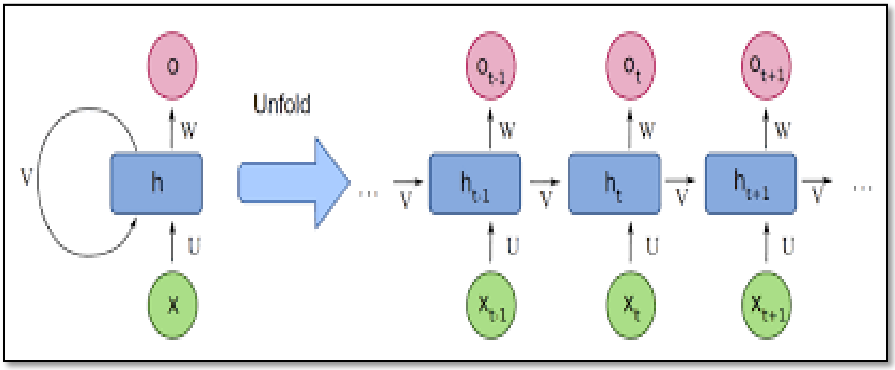


Figure 2: RNN Model

5 Algorithms

5.1 Smote

SMOTE, short for Synthetic Minority Over-sampling Technique, is like a helpful assistant in the world of machine learning. It steps in when there's an imbalance in classes, creating extra examples for the minority group by blending between existing ones. This process ensures a fairer representation in the model, reducing any bias toward the majority. So, using SMOTE is a bit like adding a special ingredient to a recipe – you need to do it thoughtfully for the best results.

When working with SMOTE, it's crucial to really think through the details, like how many nearby neighbours to consider and how much oversampling to apply. In the real world, SMOTE comes in handy in areas such as catching fraud, making medical diagnoses, and sorting through text, especially when dealing with datasets that are a bit lopsided. Think of SMOTE like a behind-the-scenes hero – it's like the wizard making sure datasets are fair and balanced, which in turn boosts how well our machine learning models perform overall.

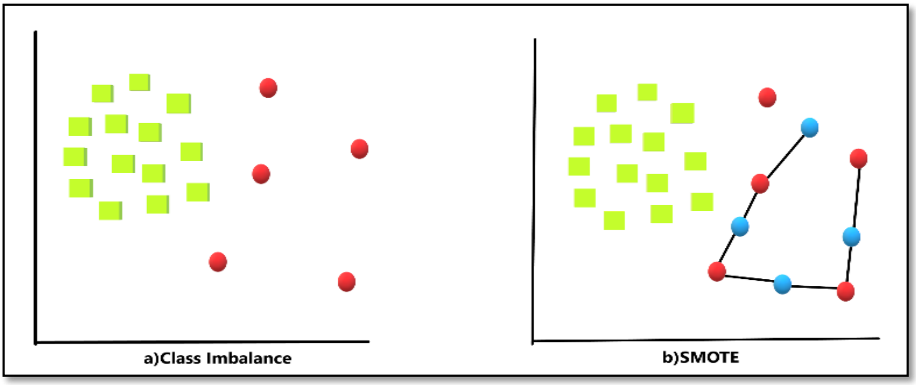


Figure 3: SMOTE for balancing an imbalance class

5.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) work a bit like your brain—they're designed to understand and remember patterns in sequences of data, just as we recognize recurring patterns in our daily experiences. What sets RNNs apart from traditional feed forward neural networks is their ability They're great at remembering past information, making them perfect for tasks like understanding time series data, language, and recognizing speech.

Recurrent Neural Networks (RNNs) are like storytellers—they analyze data one piece at a time, just like reading a book page by page. But what makes them special is their memory—they remember what happened in the story before, which helps them understand how everything fits together. It's like they're living in the moment while also keeping track of the past, which lets them understand the flow and context of the story.

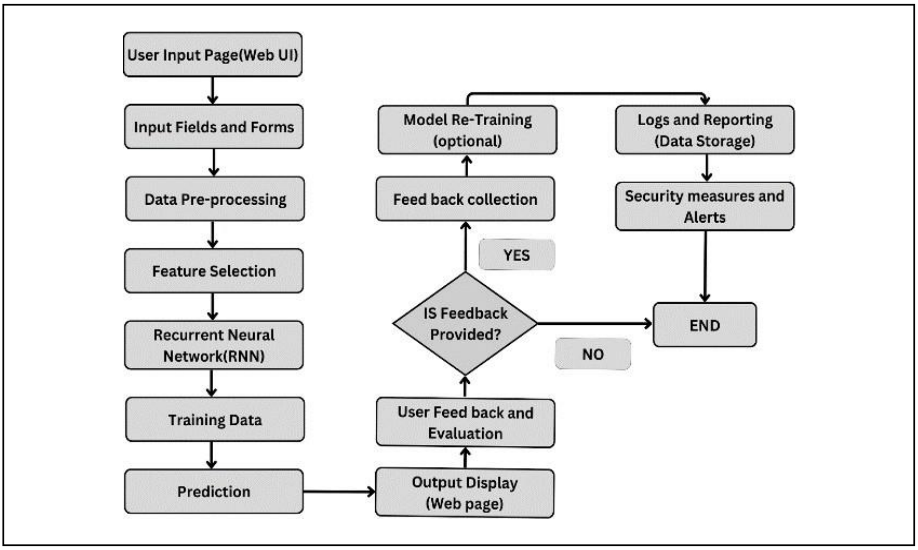

User Input Page(Web UI)
Input Fields and Forms
Data Pre-processing
Feature Selection
Recurrent Neural Network(RNN)
Training Data
Prediction
Output Display (Web page)
User Feed back and Evaluation
IS Feedback Provided?
YES
NO
Feed back collection
Model Re-Training (optional)
Logs and Reporting (Data Storage)
Security measures and Alerts
END

Figure 4: Process Flow

As the network processes each element in the sequence, it refreshes its hidden state, preserving details about the patterns and structures it has come across until now**.** However, think of traditional RNNs as learners with a hard time grasping long-distance relationships. To overcome this, smarter versions like Long Short-Term Memory (LSTM) and Gated Recurrent Unit joined the scene.

5.3 Decision Tree Classifier

To classify U2R (User to Root) and R2L (Remote to Local) attacks using a decision tree classifier, we need a well-prepared dataset with labelled instances of these attacks and relevant features. First things first, we picked up a dataset called KDD 99 from the UCI repository. Then, we put in the effort to clean, transform, and sort out the data, making sure everything was in order. We even used SMOTE to balance things out.

Once we've sorted our data into training and testing sets, we turn to Python's scikit-learn library to craft a savvy decision tree classifier. After a bit of training, we put our creation to the test, scrutinizing its performance using key metrics like accuracy, precision, recall, F1-score, and a confusion matrix on the test set. Our focus? Understanding R2L and U2R attacks, honing in on specific features like TCP destination port number, TCP window size, and IP length. What makes it even more intriguing? We unleash the ID3 algorithm, letting it weave its magic to automatically generate decision trees tailored for R2L and U2R attacks.

This sleuth learns the ropes by studying a slew of labeled examples, becoming adept at spotting normal behavior and various types of attacks. What's neat is that this investigator is an open book; it lays out its methodology, making it easier for cyber security experts to follow along. But make no mistake, being a detective isn't a walk in the park—there's the danger of getting too caught up in the details (overfitting) and the constant need for solid, trustworthy evidence (high-quality data). Yet, in the realm of cyber security, these Decision Tree detectives are indispensable guardians, keeping computer systems and networks safe and secure.

Imagine you're a cyber-detective, and your job is to figure out if there are sneaky attacks happening in a big dataset. So, you start by cleaning up thedata, making sure it's in good shape. You then look at the features—like clues that might tell you if it's a User-to-Root (U2R) attack, a Remote-to- Local (R2L) attack, or just normal stuff. After that, you split your dataset into parts for learning and testing, kind of like studying for an exam. Using a smart tool called a Decision Tree, you teach it what you've learned from the training set. Now comes the exciting part—you test it on new data to see how well it learned. If it's not perfect, no worries; you tweak it a bit, like adjusting your detective skills. Once you're satisfied, you put your detective to work in the real world, keeping an eye out for any new tricks the bad guys might try. And don't forget to check in with your detective regularly, updating their knowledge so they can stay sharp and catch those cyber villains.

Decision tree classifier is required in order to classify the attacks (R2L, U2R) as follows:

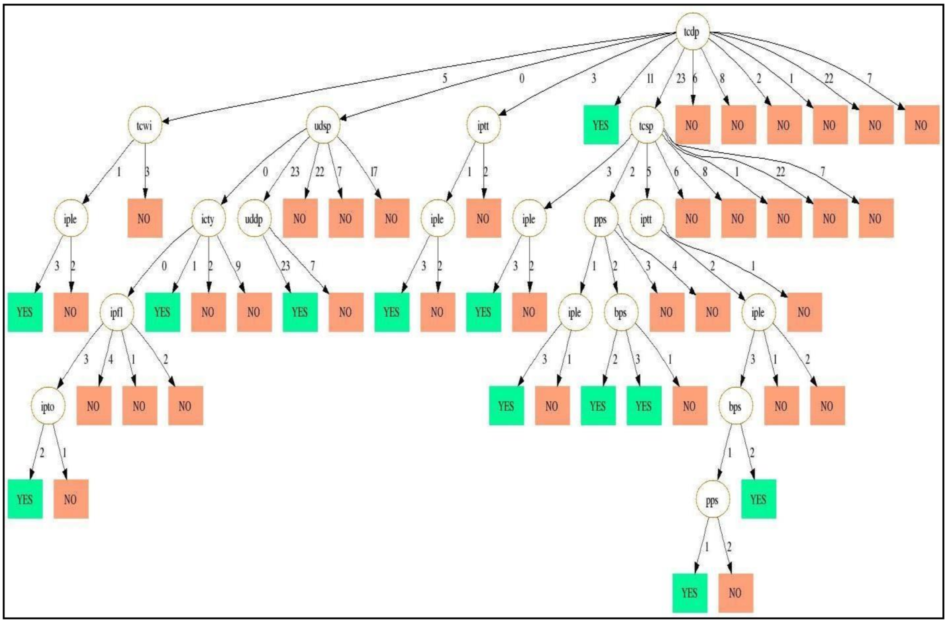


Figure 5: Decision Tree for U2R attack

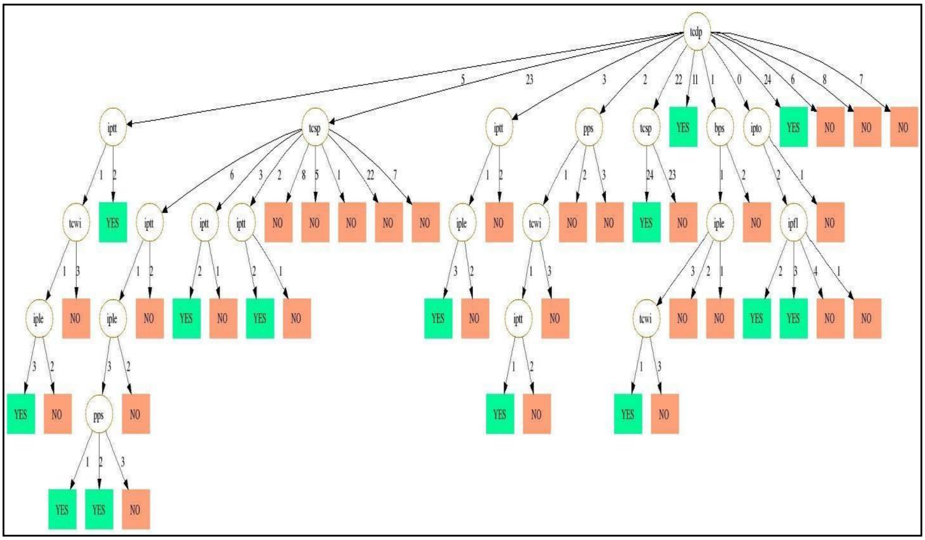


Figure 6: Decision Tree for R2L attack

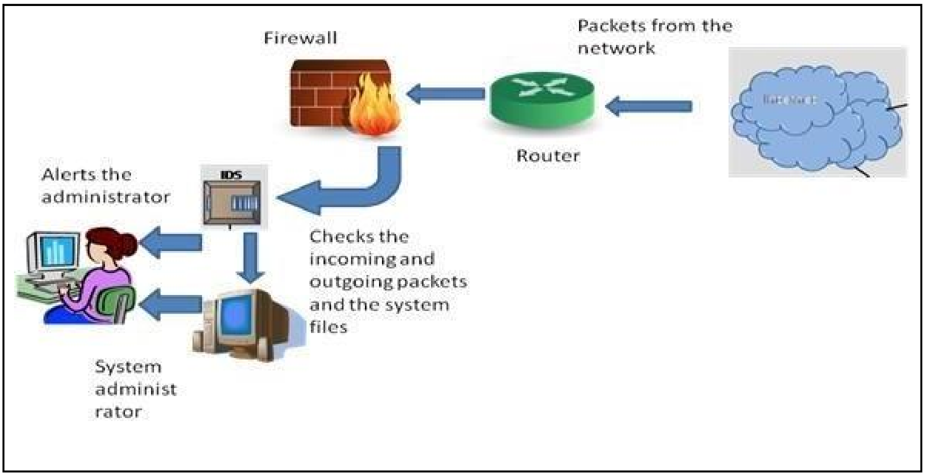
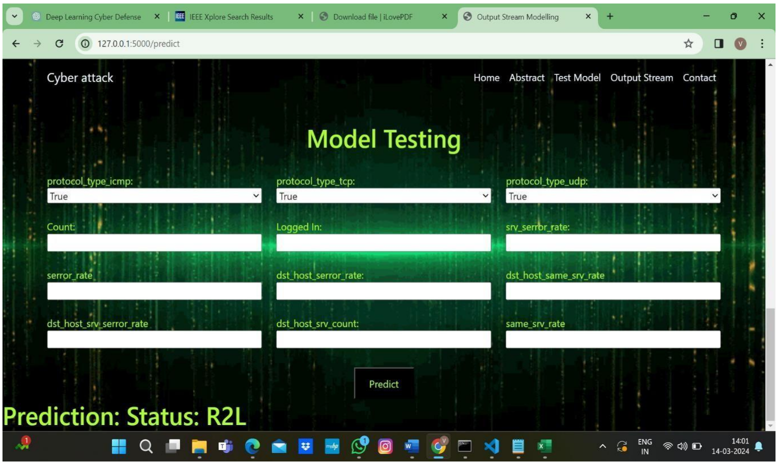


Figure 7: Architecture Diagram

6 Results



Model Testing

protocol_type_icmp: True

protocol_type_tcp: True

protocol_type_udp: True

Count:

Logged In:

srv_serror_rate:

serror_rate

dst_host_serror_rate:

dst_host_same_srv_rate

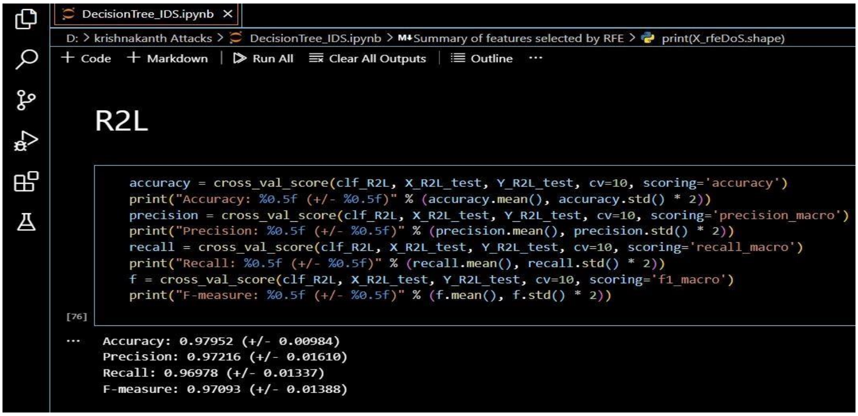
dst_host_srv_serror_rate

dst_host_srv_count:

same_srv_rate

Prediction: Status: R2L

Figure 8: web interface for users to enter input values to predict and classify the attack.

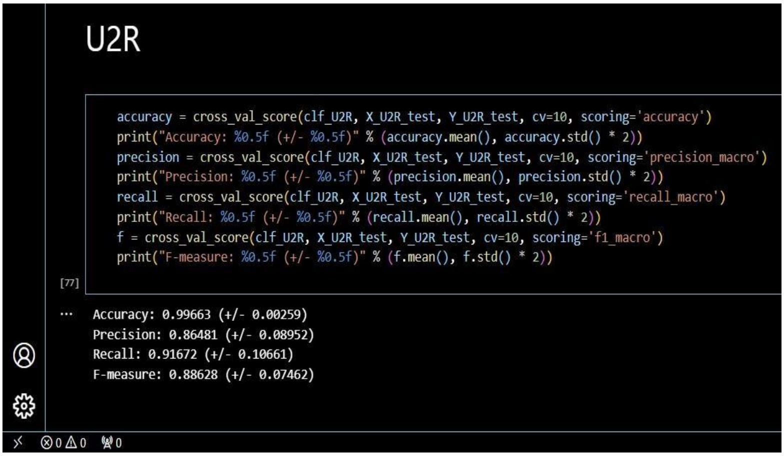

DecisionTree_IDS.ipynb
D: > krishnakanth Attacks > DecisionTree_IDS.ipynb > M↓Summary of features selected by RFE > print(X_rfeDoS.shape)
+ Code
+ Markdown
Run All
Clear All Outputs
Outline
[76]

R2L

```
accuracy = cross_val_score(clf_R2L, X_R2L_test, Y_R2L_test, cv=10, scoring='accuracy')
print("Accuracy: %0.5f (+/- %0.5f)" % (accuracy.mean(), accuracy.std() * 2))
precision = cross_val_score(clf_R2L, X_R2L_test, Y_R2L_test, cv=10, scoring='precision_macro')
print("Precision: %0.5f (+/- %0.5f)" % (precision.mean(), precision.std() * 2))
recall = cross_val_score(clf_R2L, X_R2L_test, Y_R2L_test, cv=10, scoring='recall_macro')
print("Recall: %0.5f (+/- %0.5f)" % (recall.mean(), recall.std() * 2))
f = cross_val_score(clf_R2L, X_R2L_test, Y_R2L_test, cv=10, scoring='f1_macro')
print("F-measure: %0.5f (+/- %0.5f)" % (f.mean(), f.std() * 2))
```

```
Accuracy: 0.97952 (+/- 0.00984)
Precision: 0.97216 (+/- 0.01610)
Recall: 0.96978 (+/- 0.01337)
F-measure: 0.97093 (+/- 0.01388)
```

Figure 9: Accuracy for prediction of R2L attack using decision tree classifier.



```
accuracy = cross_val_score(clf_U2R, X_U2R_test, Y_U2R_test, cv=10, scoring='accuracy')
print("Accuracy: %0.5f (+/- %0.5f)" % (accuracy.mean(), accuracy.std() * 2))
precision = cross_val_score(clf_U2R, X_U2R_test, Y_U2R_test, cv=10, scoring='precision_macro')
print("Precision: %0.5f (+/- %0.5f)" % (precision.mean(), precision.std() * 2))
recall = cross_val_score(clf_U2R, X_U2R_test, Y_U2R_test, cv=10, scoring='recall_macro')
print("Recall: %0.5f (+/- %0.5f)" % (recall.mean(), recall.std() * 2))
f = cross_val_score(clf_U2R, X_U2R_test, Y_U2R_test, cv=10, scoring='f1_macro')
print("F-measure: %0.5f (+/- %0.5f)" % (f.mean(), f.std() * 2))
```

```
Accuracy: 0.99663 (+/- 0.00259)
Precision: 0.86481 (+/- 0.08952)
Recall: 0.91672 (+/- 0.10661)
F-measure: 0.88628 (+/- 0.07462)
```

Figure 10: Accuracy for prediction of U2R attack using decision tree classifier

EXISTING SYSTEM	PROPOSED SYSTEM
1. It is not User friendly.	1.The Application is User friendly by providing webinterface.
2. CNNs typically require fixed-size input.	2. RNNs can handle sequences of variable length.
3.CNN's have Higher Computational cost andcan't be implemented in all datasets.	3.RNN can predict results with less time consumptionand also provides High Output Efficiency.
4. CNN may struggle to remember each datathroughout the time as they don't have built-in memory mechanism.	4. An RNN model is modeled to remember eachinformation throughout the time which is very help full in any time series predictor.

Table 1: Comparative study

7 Conclusion and Future Study

Predicting and classifying cyber hacking breaches using deep learning is a highly promising and crucial advancement within the boundary of cyber security. Deep learning techniques, particularly neural networks, offer numerous advantages in handling the complexities of cyber security.

They provide improvement in accuracy, as these models excel in extracting intricate features and patterns from extensive and diverse data sets.

The main advantage of our project is not only predicting the cyber hacking breaches but also classify which attack it is suffering from. As technology continues to advance, cyber security professionals will play a vital role in safeguarding digital assets, privacy, and critical infrastructure, making it a field with sustained growth and continuous innovation.

Cyber-attacks prediction should be improved in order to predict new attacks by recognizing the patterns effectively. Because there is a high-level chance for Cyber threats to improve their damage level at higher rates. In future, there should be an effective prediction and classification model such that it can predict, classify and guide the users by taking appropriate measures to reduce the upcoming damage indeed getting worse in terms of frequency.

8 References

1.J. -H. Lee, J. -H. Lee, S. -G. Sohn, J. -H. Ryu and T. -M. Chung, "Effective Value of Decision Tree with KDD 99 Intrusion Detection Datasets for Intrusion Detection System," 2008 10th International Conference on Advanced Communication Technology, Gangwon, Korea (South), 2008, pp. 1170-1175, doi: 10.1109/ICACT.2008.4493974.
2. M. Feily, A. Shahrestani and S. Ramadass, "A Survey of Botnet and Botnet Detection," 2009 Third International Conference on Emerging Security Information, Systems and Technologies, Athens, Greece, 2009, pp. 268-273, doi: 10.1109/SECURWARE.2009.48.
3. S. Suthaharan, "An iterative ellipsoid-based anomaly detection technique for intrusion detection systems," 2012 Proceedings of IEEE Southeastcon, Orlando, FL, USA, 2012, pp. 1-6, doi: 10.1109/SECon.2012.6196956.
4. Polamuri, S.R., Srinnivas, K. & Mohan, A.K. Prediction of stock price growth for novel greedy heuristic optimized multi-instances quantitative (NGHOMQ). Int J Syst Assur Eng Manag 14, 353–366 (2023). https://doi.org/10.1007/s13198-022-01801-3
5. G. ZHU, H. YUAN, Y. ZHUANG, Y. GUO, X. ZHANG and S. QIU, "Research on network intrusion detection method of power system based on random forest algorithm," 2021 13th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Beihai, China, 2021, pp. 374-379, doi: 10.1109/ICMTMA52658.2021.00087.
6. Kumar, Voruganti Naresh, U. Sivaji, Gunipati Kanishka, B. Rupa Devi, A. Suresh, K. Reddy Madhavi, and Syed Thouheed Ahmed. "A Framework For Tweet Classification And Analysis On Social Media Platform Using Federated Learning." Malaysian Journal of Computer Science (2023): 90-98.
7.A. Kiran, S. W. Prakash, B. A. Kumar, Likhitha, T. Sameeratmaja and U. S. S. R. Charan, "Intrusion Detection System Using Machine Learning," 2023 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India,2023, pp. 1-4, doi: 10.1109/ICCCI56745.2023.10128363.