IMPROVING CYBER THREAT DETECTION USING AI

Ashwath D Padur, Chethan S and G S Sudeep
Department of Computer Science Engineering
Don Bosco Institute of Technology, Bengaluru, India
ashwath29may@gmail.com,
chethans20012017@gmail.com,
sudeepkrishna187@gmail.com

Abstract: Security awareness has been a popular topic in the last few years for both information systems researchers and organizations. News broadcasts has brought attention to the increase in cyber- attacks, with these reports noting that a significant number of these breaches have been caused by human error, linked to employee's lack of engagement with their organizations security policies and awareness campaigns. Whilst there is existing research in human factors and the barriers of security behaviour's effect on cyber security awareness; in practice we know very little about how employees can be motivated to engage in cyber security awareness programs. The software's like Pycharm and Jupyter notebook is used to build the project, "Improving Cyber Threat Detection Using AI". Building a project which shows the clear difference between the different AI models by showing their attributes like, accuracy and precision and provide a pictorial representation of these attributes.

Background: AI-based cyber threat detection systems use machine learning algorithms and other AI techniques to analyse large amounts of data and detect anomalous behaviour or patterns that may indicate a security threat. These systems can learn from past attacks and adapt to new threats, making them more effective at detecting and responding to cyber threats in real-time.AI-based cyber threat detection systems can be used in a variety of settings, including government agencies, financial institutions, healthcare organizations, and corporations. They can help organizations detect and respond to cyber threats more quickly and effectively, reducing the risk of data breaches, financial losses, and reputational damage. However, AI-based cyber threat detection systems are not fool proof and require ongoing monitoring and updating to remain effective. Organizations must also balance the benefits of AI-based threat detection with the potential risks, such as false positives and the potential for AI algorithms to be compromised or manipulated by cyber attackers.

Objective: To display the accuracy, precision and fmeasure of each AI model. To compare the differences of accuracy, precision and fmeasure of each model. To develop a model by integrating two models with the highest consistent accuracy, precision and fmeasure compared to all different model. Compare the accuracy, precision and fmeasure of all the AI and create a model with highest accuracy, precision and fmeasure by combining two models.

Methods: The datasets are extracted from kaggle to train and test the accuracy, precision, fmeasure and recall of KNN, SVM, Random Forest, CNN, Decision Tree, LSTM and Naïve Bayes AI models. And a new model is created by integrating PSO and SVM model. SVM works by finding hyper plane that maximally separates the data points into different classes based on their feature values. PSO works by iteratively updating the preposition and velocity of the swarm particles. SVM with PSO is a hybrid algorithm that combines the power of SVM for classification with the optimization capabilities of PSO for hyper plane parameter tuning. The AI models are trained using the labelled datasets which increases the training efficiency which in turn increases the accuracy of the AI model

Results: SVM with PSO had the highest precision, accuracy and fmeasure. SVM with PSO had increased the precision, accuracy and fmeasure by 10% to 20% when compared to other AI models and was a close second to Random Forest AI Model in recall value.

Conclusions: Naïve Bayes, Random Forest, KNN, CNN, LSTM, SVM AI models is used and implemented to display their accuracy and precision. A new model is created by integrating SVM with PSO models. A graph is shown to display the comparison of accuracy and precision between the different existing models and the newly created model. SVM with PSO can improve the performance of SVM by reducing over fitting and increasing the generalisation ability of the model, as well as reducing the computational cost of hyper-parameter tuning.

Keywords: Artificial Intelligence, K-Nearest Neighbour, Convolution Neural Network, Long Short-Term Memory, Particle Swarm Optimization.

1. Introduction

Cyber security is defined as a set of process, human behavior and systems that help safeguard electronic resources. Cyber security is a fast-evolving field over the last decade. As we rely more and more on technology, the need for better security also increases. Cyber security is the application of technologies, processes, and controls to protect systems, networks, programs, devices and data from cyber-attacks. AI models can be used to train the system to detect malicious threats to the system and improve the security. Most cybercrime is committed by cybercriminals or hackers who want to make money. However, occasionally cybercrime aims to damage computers or networks for reasons other than profit. These could be potential or personal. The basic goal of cyber security is to protect the confidentiality of all business data from external and internal threats as well as disruptions brought on by natural disasters. Cyber security professionals are always busy outsmarting black hat hackers, patching vulnerabilities and analyzing the risk of an organization. Tackling such attacks in an ever- advancing industry only comes with continuous study and thorough research.

2. Literature Survey

[1] Sherali Zeadally, et.al "Harnessing Artificial intelligence Capabilities to improve Cyber Security"

As humans are depending more and more on machines and software's, securing these is essential. Advanced cyber-attacks along with common attacks such as Man in the middle attack, Denial-of-service attack, etc. are emerging every day and the losses encurred due to these attacks are enormous. Manual (non-AI) cyber security techniques such as Rate Control where attacks such as Denial-of-service or Distributed Denial-of-service by reducing the volume of incoming network traffic, Signature-based intrusion detection where a database containing familiar digital signatures are compared with signatures in the traffic, etc. are difficult to implement manually and also prove ineffective against a slightly modified attack. This is where Artificial Intelligence proves to be effective. AI techniques such as Machine Learning, Decision trees, Support Vector Machines, Artificial Neural network are used to detect and prevent cyber-attacks through various ways such as classifying digital signatures using KNN algorithm, or detecting malicious codes using Long-Short-Term memory. Drawbacks include high implementation costs, computational cost and training the model and sometimes low precision and accuracy. But these are still highly effective and efficient compared to traditional non-AI techniques. Hence, we are constantly searching for new ways to improve the accuracy and precision of the AI models.

Advantages:

- Detection of cyber threats are carried out more efficiently than traditional methods.
- Using the natural language processing feature in AI, security professionals can detect the origin of a cyber-attack.
- Modern firewalls will have built-in machine learning technology that will easily detect a usual pattern in the network traffic and remove it if considered malicious

Disadvantages:

- Difficulty in getting training sets to train AI models.
- Sometimes accuracy and precision of output from AI models will be low
- High computational and processing costs.

[2] Nicola Capuano and et.al "Explainable Artificial Intelligence in Cyber Security"

Although Artificial Intelligence is widely applied in cyber security because of its effectiveness, its application suffers from the opacity of complex internal mechanisms and doesn't satisfy by design the principles of Explainable Artificial Intelligence (XAI). The lack of transparency further exacerbates the problem in the field of Cyber Security because entrusting crucial decisions to a system that cannot explain itself presents obvious dangers. The absence of transparency undermines confidence. Security practitioners may hesitate to trust the systems if they do not understand how crucial decisions are made. XAI aims to explain the logic behind the output of an AI model. The transparency can substantially improve Cyber Security practices but it may also facilitate new attacks on the AI applications since it will also be

Explainable to the attacker, which may pose severe security threats but the pros outweigh the cons mitigating the risks of AI adoption. Hence use of AI models such as Convolution Neural Network and Support Vector machine is encouraged due to its transparency.

Advantages:

- XAI introduces transparency in AI models which encourages people to adopt them.
- Features in AI models can be modified if the complete working of that model is known.

Disadvantages:

- The transparency produced by XAI is exploited by cyber criminals to find loopholes in the model and exploit it.
- Criminals can easily predict the logic and output of a model and exploit it

[3] Pooja S and et.al "Developer's Roadmap to Design Software Vulnerability Detection Model Using Different AI Approaches"

The key requirement for developing an AI based vulnerability detector model from a developer perspective is to identify which AI model to adopt, availability of labelled dataset, how to represent essential features. Software vulnerability detection can be modeled as Natural Language Processing (NLP) problem with source code treated as texts. The vulnerability detection models aim for binary classification i.e. categorizing input code as vulnerable or secure code; or as a multiclass classification, additionally classifying into particular type of vulnerability. Considering all the above requirements, using AI models such as Long-Short-term Memory, Convolutional Neural Network (CNN), K-Neighbors Classifier are encouraged since the datasets are easily available, data can be classified into groups easily and are easily represented.

Advantages:

- AI models can be trained easily as the datasets are abundantly available.
- Datasets can be classified and represented using suitable approaches.

Disadvantages:

- Models that meet the above requirements is not suitable to detect all the vulnerabilities and attacks.
- As datasets and its representation is easily available, criminals can exploit the software and attack it as they will be able to predict the working of the cyberThreat detection system.

[4] Hatma Suryotrisongko and et.all "Robust Botnet DGA Detection: Blending XAI and OSINT for Cyber Threat Intelligence Sharing"

Domain generation algorithm (DGA) based botnets are difficult to detect using cyber threat detection systems based on block lists. Artificial intelligence /machine learning based Cyber Thread detection systems are required. Cyber Threat detection systems can be expanded to produce improved accuracy and precision and gain trust by the explain ability of the model outputs by blending explainable Artificial Intelligence (XAI) and open-source intelligence (OSINT). The models such as Random Forest provides better robustness against DGA adversarial attacks such as
CharBOT and MaskDGA compared with character based deep learning models such as CMU and MIT. These models can be used in combination of various features such as Char Length and entropy to improve the accuracy of the AI model.

Advantages

- Blending XAI and OSINT with each other will produce better detection for some attacks such as DGA based botnet traffic.
- Extending the model with various features produces improved accuracy in output

Disadvantages

- Higher computational and implementation costs.
- Extending the model with various features requires complex design and coding for proper working of the model.

[5] Kamran Shaukat and et.all "Cyber Threat Detection Using Machine Learning Techniques: A Performance Evaluation Perspective"

As the need for cyber security is increasing day by day, researchers are finding new ways to implement AI/ML models into cyber security. There is no machine learning technique that is not vulnerable to cyber-attacks. Every machine learning technique is still struggling to keep a pace with continuously upgrading cybercrimes. But compared to different classifiers and AI models, the convolutional neural network (CNN) classifier is an underused classifier, and it could have brought vast advancements in cyber security if it was used to its full potential. Hence the CNN has wide scope in cyber security as a classifier. Different models are suitable for different attacks such as Support Vector Machine and Decision tree models are suitable for intrusion detection. Hence no model is perfectly suitable for every attack and no model is perfectly accurate.

3. Methodology

The methodology for improving cyber threat detection using AI generally involves the following steps:

Data collection: The first step is to gather data from various sources such as network traffic logs, system logs, and security event logs.

Data pre-processing: The collected data is pre-processed by removing irrelevant information, cleaning up the data, and converting it into a suitable format for analysis.

Feature extraction: The pre-processed data is analyzed, and features that can be used to identify potential threats are extracted. These features can include network traffic patterns, system logs, user behavior, and other indicators of compromise.

Algorithm selection: Different machine learning algorithms are evaluated to determine which one is best suited to detect the specific types of threats identified in the data.

Model training: The selected algorithm is trained on a labeled dataset to learn patterns and identify anomalies that indicate potential threats.

Model validation: The performance of the trained model is evaluated using a separate dataset to ensure that it can accurately detect threats without generating too many false positives or false negatives.

Model deployment: Once the model has been validated, it can be deployed in a production environment to monitor network activity and detect potential threats.

Model updating: The model is continually updated to incorporate new threat intelligence and adapt to changes in the threat landscape.

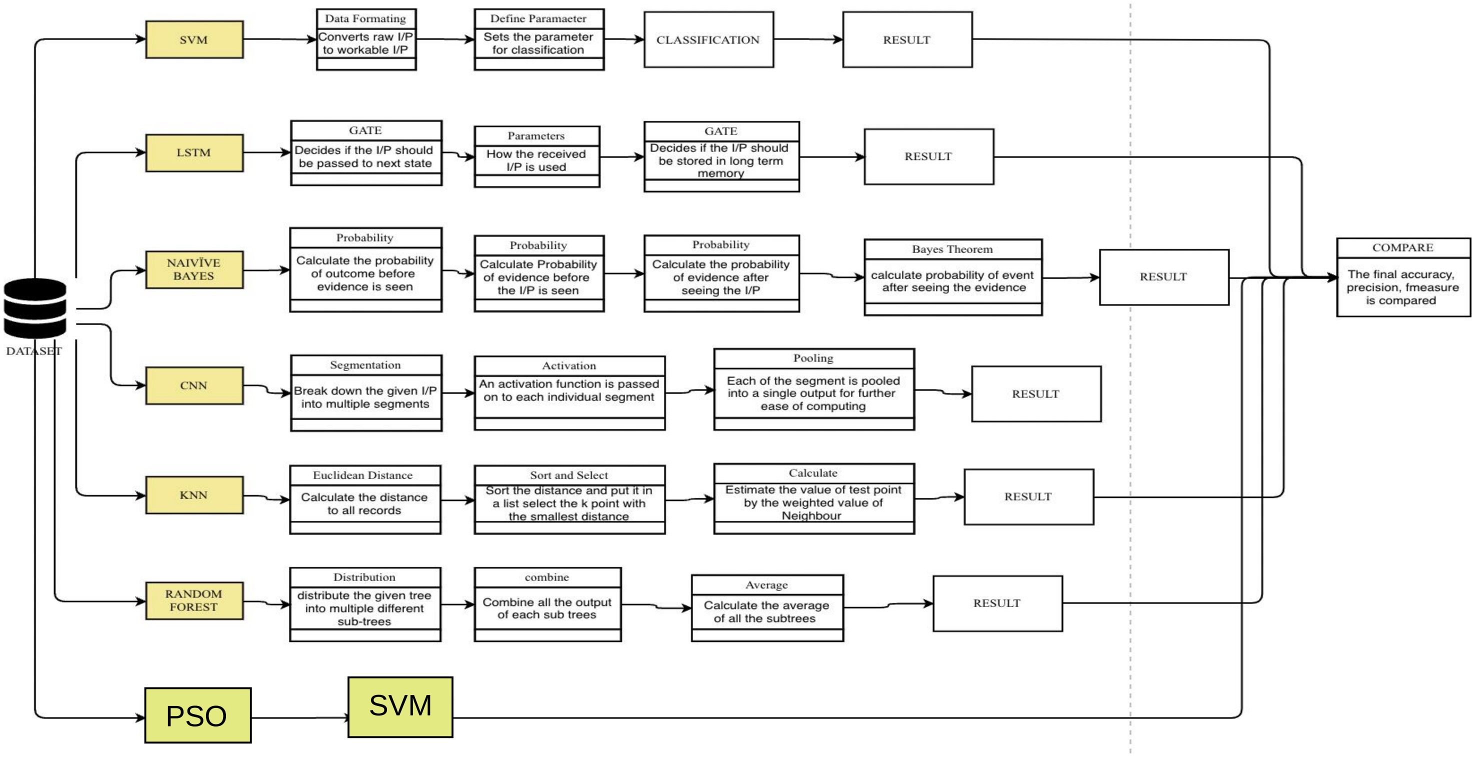

DATASET
SVM
Data Formating
Converts raw I/P to workable I/P
Define Paramaeter
Sets the parameter for classification
CLASSIFICATION
RESULT
LSTM
GATE
Decides if the I/P should be passed to next state
Parameters
How the received I/P is used
GATE
Decides if the I/P should be stored in long term memory
RESULT
NAIVÏVE BAYES
Probability
Calculate the probability of outcome before evidence is seen
Probability
Calculate Probability of evidence before the I/P is seen
Probability
Calculate the probability of evidence after seeing the I/P
Bayes Theorem
calculate probability of event after seeing the evidence
RESULT
CNN
Segmentation
Break down the given I/P into multiple segments
Activation
An activation function is passed on to each individual segment
Pooling
Each of the segment is pooled into a single output for further ease of computing
RESULT
KNN
Euclidean Distance
Calculate the distance to all records
Sort and Select
Sort the distance and put it in a list select the k point with the smallest distance
Calculate
Estimate the value of test point by the weighted value of Neighbour
RESULT
RANDOM FOREST
Distribution
distribute the given tree into multiple different sub-trees
combine
Combine all the output of each sub trees
Average
Calculate the average of all the subtrees
RESULT
PSO
SVM
COMPARE
The final accuracy, precision, fmeasure is compared

Figure 1: Data Flow Diagram of Cyber Threat Detection using AI

3.1 Data Collection

The data is collected from Kaggel as it provides the dataset which is suitable for most of the models and is OpenSource. The dataset contains 42 attributes of which we have considered all of them. Kaggel provides a reliable dataset to test and train the AI model.
Below is the list of highlighted 42 attributes used in this research paper:

Duration
Protocol type
Service
Flag
Src bytes
Dst bytes
Land
Wrong fragment
Urgent
Hot
Diff srv rate
Dst host srv count
Dst host srv diff host rate
Dst host rerror rate

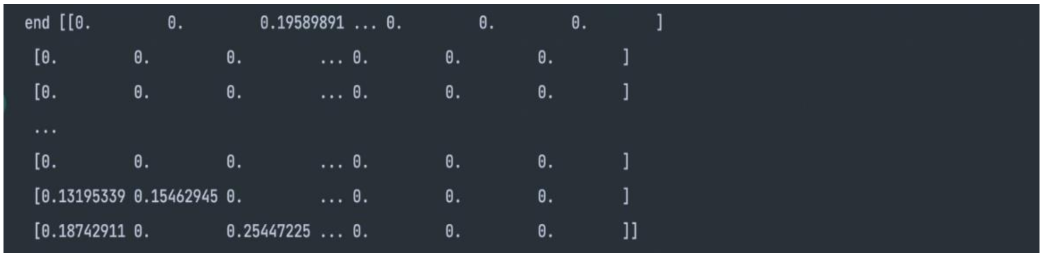
Num failed logins
Logged in
Num compromised
Root shell
Su attempted
Num root
Num file creations
Num shells
Num access files
Same srv rate
Srv diff host rate
Dst host same srv rate
Dst host serror rate
Dst host srv rerror rate

Num outbound cmds
Is host login
Is guest login
Count
Srv count
Serror rate
Srv serror rate
Rerror rate
Srv rerror rate
Dst host count
Dst host diff srv rate
Dst host same src port rate
Dst host srv serror rate
labels

Table 2: Attributes of the Improving Cyber Threat Detection using AI

3.2 Data Preprocessing

Data preprocessing is the process of preparing raw data for analysis by cleaning, transforming, and reducing its size or complexity. This step is essential to ensure the quality and accuracy of the data used for analysis, especially in machine learning applications. Data cleaning involves identifying and correcting errors or inconsistencies, while data transformation involves converting the data into a suitable format for analysis. Data reduction and integration are also important steps to make the data more manageable and relevant. Overall, data preprocessing is crucial to ensure the validity and effectiveness of any data analysis or modeling. TF-IDF (Term Frequency-Inverse Document Frequency) is a statistical algorithm used to evaluate the relevance of a word in a text document. It measures the frequency of a term in a document (TF) and the rarity of the term across all documents (IDF) to calculate a weight for the term. The resulting TF-IDF score indicates the importance of the term in the document and can be used to identify relevant keywords or phrases.

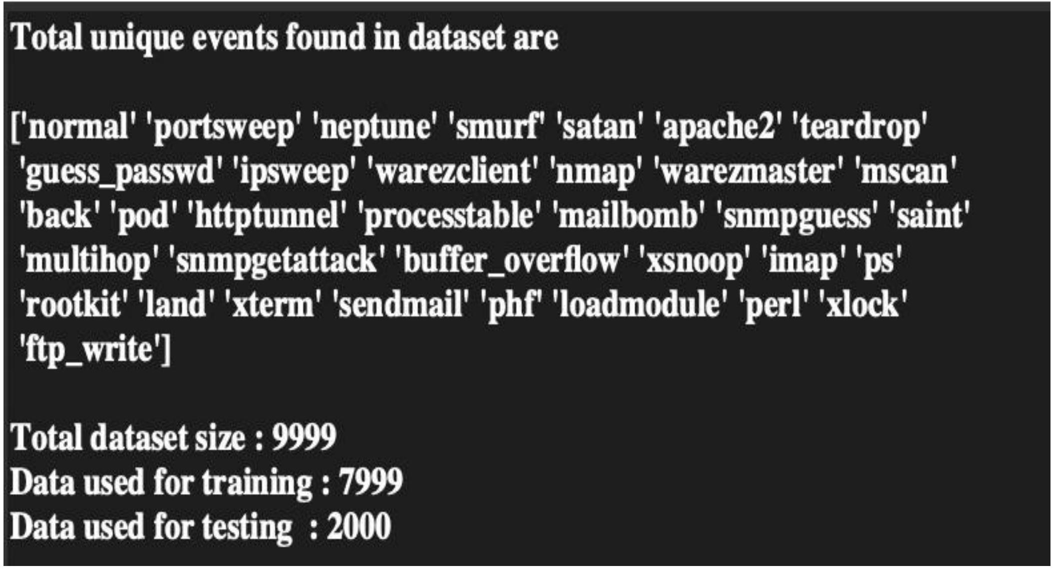


```
end [[0.         0.         0.19589891 ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 ...
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.13195339 0.15462945 0.         ... 0.         0.         0.        ]
 [0.18742911 0.         0.25447225 ... 0.         0.         0.        ]]
```

Figure 2: Pre Processing using TF-IDF Algorithm

3.3 Event Vector Generation

Event vector generation is a process of representing events or logs in a machine-readable format using numerical vectors. This process involves converting each event into a feature vector that contains information about the event, such as its type, severity, source, and timestamp. The feature vector can also include other relevant information, such as user behavior or network traffic patterns. Event vector generation is often used in anomaly detection and other cyber security applications to analyze large amounts of event data and identify potential threats or security incidents. The resulting vectors can be input into machine learning algorithms to detect anomalies or predict future events.



```
Total unique events found in dataset are

['normal' 'portsweep' 'neptune' 'smurf' 'satan' 'apache2' 'teardrop'
 'guess_passwd' 'ipsweep' 'warezclient' 'nmap' 'warezmaster' 'mscan'
 'back' 'pod' 'httptunnel' 'processtable' 'mailbomb' 'snmpguess' 'saint'
 'multihop' 'snmpgetattack' 'buffer_overflow' 'xsnoop' 'imap' 'ps'
 'rootkit' 'land' 'xterm' 'sendmail' 'phf' 'loadmodule' 'perl' 'xlock'
 'ftp_write']

Total dataset size : 9999
Data used for training : 7999
Data used for testing  : 2000
```

Figure 3: Event Vector Generation

3.4 Proposed Algorithms

3.4.1 Naïve Bayes

Naïve Bayes algorithm is a supervised learning algorithm based on Bayes theorem. It predicts the outcome based on the probability of object. Bayes theorem is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

Bayes theorem = $P(A|B) = [\ P(B|A)\ P(A)\ ] / P(B)$

$P(A|B)$ = Probability of hypothesis A on the observed event B.

$P(B|A)$ = Probability of the evidence given that the probability of a hypothesis is true.

$P(A)$ = Probability of hypothesis before observing the evidence.

$P(B)$ = Probability of evidence.

3.4.2 Random Forest

Using the given dataset, separate the given set into multiple different sets.
Using the different sets, decision trees are generated.
Usually, 100 different decision trees are generated. These decision trees generate the output.
Then the output from different trees are taken together and an average is generated which is taken as the output.

3.4.3 KNN – K Nearest Neighbor

K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most like the available categories. The category is chosen based on the nearest sorted category.

3.4.4 CNN – Convolution Neural Network

A large dataset is provided. It is then broken down into small parts. The small parts are given as input to the algorithm. Similarly, the process is repeated till the entire dataset is traversed. The data is then fed into the model and output from each layer is obtained this step is called feed forward, we then calculate the error. Later, we back propagate into the model by calculating the derivatives. This step is called Back propagation which is used to minimize the loss.This is useful to simplify complex datasets.

3.4.5 LSTM – Long Short-Term Memory

LSTM is a recurring Neural Network. LSTM predicts an output by having the output of previous input. The central role of an LSTM model is held by a memory cell known as a ‘cell state’ that maintains its state over time. The required data can be stored in the long-term memory and the unwanted data are discarded. The passing of the info in each state is dependent on the sigmoid function. '0' indicates no info should be passed and '1' indicates all the data should be passed.

3.4.6 SVM – Support Vector Machine

All the characteristics are differentiated into different categories. A Hyper plane is drawn between the categories and the margin is drawn between the 2 closest nodes on either side of the hyper plane. The two closest nodes are referred to as support nodes. After training the model, the classification can be done by placing choosing the correct plane.

3.4.7 Decision tree

Decision trees are a type of supervised machine learning algorithm used for classification and regression tasks. A decision tree consists of a root node, internal nodes, and leaf nodes. Each internal node represents a decision based on a feature value, and each leaf node represents a class label or numerical value. Decision trees are easy to interpret and can handle both categorical and numerical data. They can also handle missing values and outliers by assigning them to the most common class or value at each node. Decision trees can suffer from over fitting if the tree is too deep or if there is too much noise in the data. They also tend to be unstable, meaning that small changes in the data can result in large changes in the tree structure. To address these issues, ensemble methods like Random Forest and Gradient Boosting can be used to combine multiple decision trees into a more robust model. Pruning techniques like post-pruning and pre-pruning can also be used to reduce over fitting.

3.4.8 SVM with PSO

SVM works by finding the hyper plane that maximally separates the data points into different classes, based on their feature values. PSO works by iteratively updating the positions and velocities of a swarm of particle. SVM with PSO is a hybrid algorithm that combines the power of SVM for classification with the optimization capabilities of PSO for hyper-parameter tuning. The main goal of SVM with PSO is to find the best combination of hyper-parameters for the SVM algorithm, such as the kernel function, kernel parameters, and regularization parameter. SVM with PSO can improve the performance of SVM by reducing over fitting and increasing the generalization ability of the model, as well as reducing the computational cost of hyper-parameter tuning.

4. Result and Discussions

	Accuracy	Recall	Fmeasure	Precision
KNN	35%	7%	32%	13%
CNN	38%	27%	20%	39%
Naïve Bayes	30%	13%	53%	58%
Random Forest	86%	55%	19%	38%
SVM	37%	20%	47%	22%
Decision Tree	50%	15%	10%	19%
LSTM	72%	22%	23%	26%
SVM with PSO	95%	47%	57%	72%

Table 1.1: Comparison of all the models which includes all the 42 attributes

The above table shows the comparison of the KNN, CNN, Naïve Bayes, Random Forest, SVM, Decision Tree, LSTM and SVM with PSO. The Accuracy, Fmeasure and Precision shows the SVM with PSO model provides high percentage whereas in Recall the Random Forest provides high percentage In our project we have used all the attributes for the comparison i.e., all 42 attributes are used for better understanding of the project.

5. Conclusion and Future Enhancement

Naïve Bayes, Random Forest, KNN, CNN, LSTM, SVM AI models is used and implemented to display their accuracy and precision. A new model is created by integrating SVM with PSO models. A graph is shown to display the comparison of accuracy and precision between the different existing models and the newly created model. Future enhancements for improving threat detection using AI could focus on developing methods for model explainability, incorporating more diverse data sources, and leveraging advanced techniques such as deep learning and natural language processing. Explainability is a significant challenge for AI-based threat detection models, and developing methods for interpreting the decision-making process of these models could improve transparency and trust in the results. Incorporating more diverse data sources, such as network traffic data, social media data, or IoT data, could improve the accuracy and effectiveness of the models. Additionally, leveraging advanced techniques such as deep learning and natural language processing could enable the models to detect more complex and sophisticated threats.

References

[1] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and deep networks intrusion detection system: A taxonomy and survey," arXiv preprint arXiv: 1701.02145, 2017.

[2] G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Marchetti, "On the effectiveness of machine and deep learning for cyber security," in 2018 10th International Conference on Cyber Conflict (CyCon), 2018: IEEE, pp. 371-390.

[3] S. Sheikhi, M. Kheirabadi, and A. Bazzazi, "An Effective Model for SMS Spam Detection Using Content-based Features and Averaged Neural Network," International Journal of Engineering, vol. 33, no. 2, pp. 221-228, 2020.

[4] F. Mercaldo and A. Santone, "Deep learning for image-based mobile malware detection," Journal of Computer Virology and Hacking Techniques, pp. 1-15, 2020.