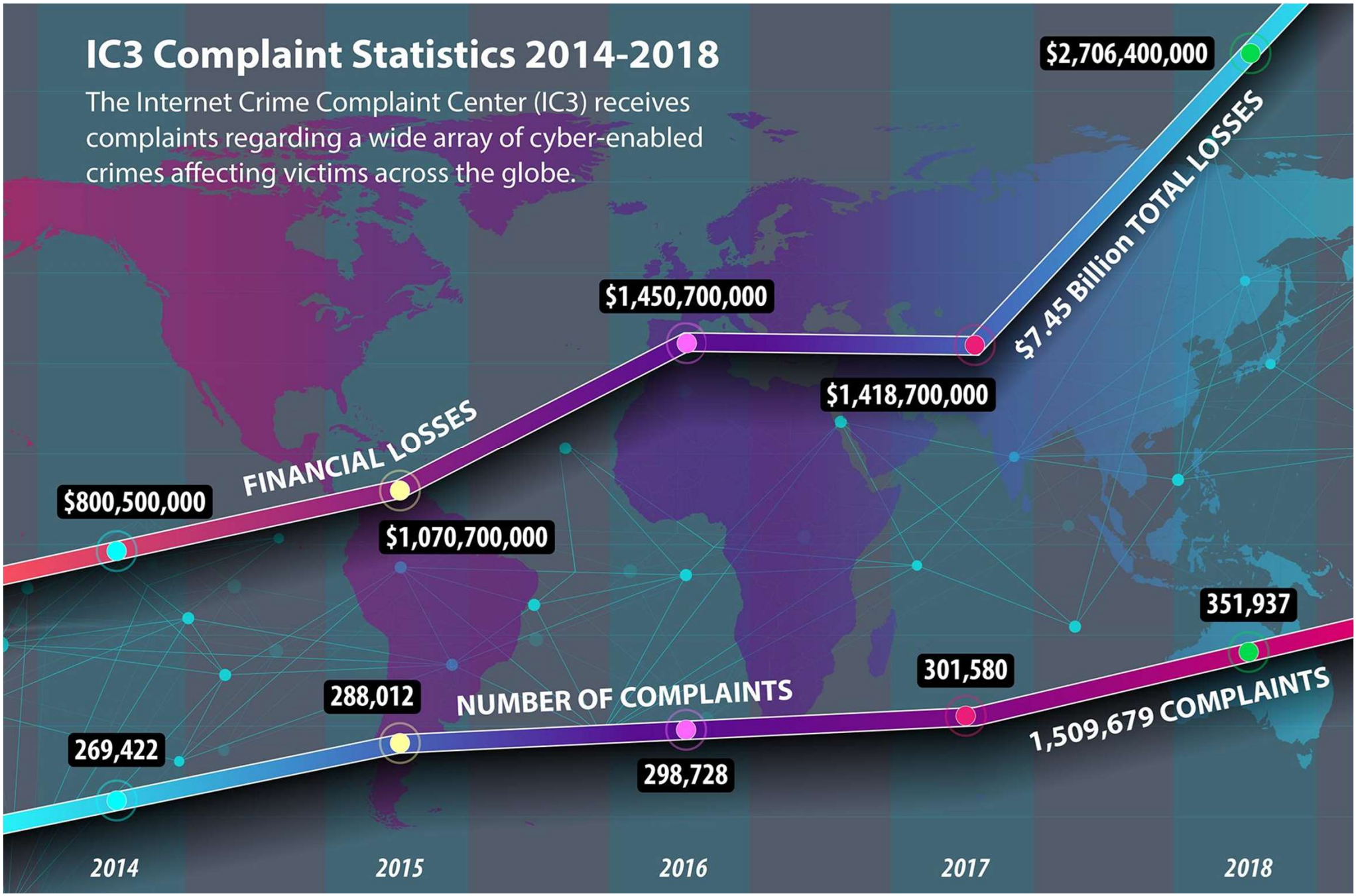
Problem and Purpose

This project focuses on the critical task of improving network intrusion detection. Network Intrusion Detection Systems (NIDS) protect computer networks by analyzing incoming network traffic and blocking packets that are identified as malicious. Current NIDS tend to be either signature based or anomaly based. Signature based NIDS compare traffic to those in a database and therefore cannot identify new types of attacks. Anomaly based approaches flag any network traffic that is unusual, resulting in many false alarms. In this project, various cutting edge artificial intelligence models were compared that have the potential to identify unseen attacks while maintaining a low false alarm rate.

Importance

Cyber attacks are rapidly increasing in frequency and severity. In March 2019, Accenture Security reported a 67% increase in security breaches over the last five years. Some of the largest security breaches have occurred over this period, impacting many millions (sometimes billions) of people. Businesses and governments across the globe spend an ever-increasing amount of funds protecting their electronic data. Contributing to the above trends and the complexity of cyber security is the daily growth in the number and type of devices making use of the internet. The number of household appliances connected to the internet, commonly referred to as the Internet of Things, provides hackers with an increasing number of opportunities to breach systems (Paul, 2019). Companies lack the necessary cybersecurity staff to defend against all possible threats. According to $(ISC)^2$, there is currently a shortage of almost 3 million cybersecurity jobs worldwide (Ackerman, 2019). The increasing complexity of attacks and scarcity of experts to defend against them are why effective automated systems have become essential.



"The statistics gathered by the FBI's Internet Crime Complaint Center (IC3) for 2018 show Internet-enabled theft, fraud, and exploitation remain pervasive and were responsible for a staggering $2.7 billion in financial losses in 2018" (Federal Bureau of Investigation, 2019).

Image taken from FBI.gov

Dataset

- Dataset contains **2,830,743 instances** of network traffic collected in 2017
- Contains network traffic information for both **benign activity** and the most **up-to-date common cyber attacks**.
- Attacks include Brute Force FTP, Brute Force SSH, DDoS, DoS, Web Attacks, Infiltration, and Botnets (see Attack Types for details)
- Over **50 GB** of data were collected over 5 days
- **84 features** were extracted from the network traffic
- Data is available in both **CSV** files with extracted features and class labels and **raw PCAPs**.

Attack Types

- **DOS/DDOS (Distributed) Denial of Service**: A denial-of-service attack floods systems, servers or networks with traffic to exhaust resources and block access. A DDOS attack is launched from many host machines that are infected by malicious software controlled by a hacker.
- **Botnet**: A botnet is a network of devices that have been infected with malicious software and under the control of a hacker for carrying out attacks (such as DOS).
- **Brute Force (SSH and FTP)**: a hacker uses code to guess all possible passwords for an administrator to gain unwarranted entrance to a system.
- **Infiltration**: Infiltration attacks exploit vulnerable software to open a backdoor on a victim's computer and carry out a variety of attacks including IP sweep and full port scans.
- **Port Scan**: A hacker uses specialized software to scan for vulnerable ports in a system that can be exploited with malicious software.
- **Web Attack (SQL Injection, Brute Force, XSS)**: an attacker inserts malicious code into a server and forces it to reveal information it normally would not.

Data Preprocessing

Cleaning

- Eight features that contained no variance were removed
- Two features that contained missing values were removed

Encoding

- All continuous features were standardized, adjusting means to zero and standard deviations to one
- Categorical features were “one hot” encoded to allow them to be input as numeric values without falsely implying an order between categories

Feature
0
1
2

OHE

Feature0	Feature1	Feature2
1	0	0
0	1	0
0	0	1

Class Grouping

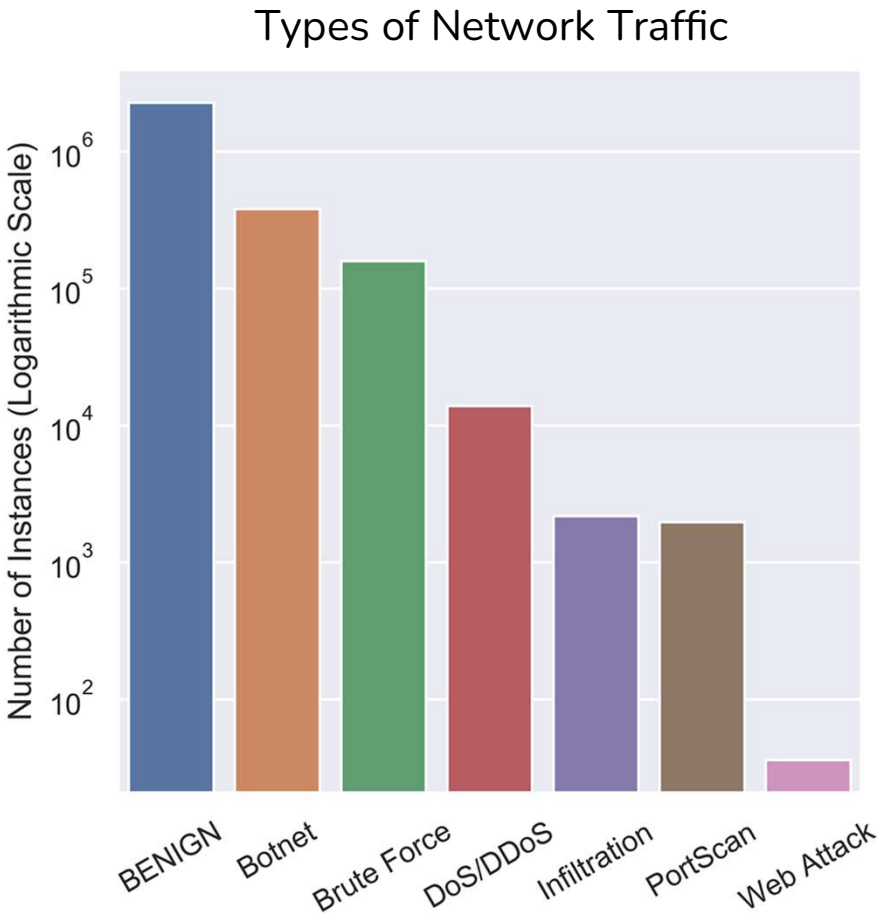
- For multiclass identification, attacks that were similar were grouped into larger categories (e.g. different DoS tools were grouped into a single DoS/DDoS category)

Train/Validation/Test Split

- The dataset was split into a 60% training set, 20% validation set, and 20% test set
- The training set is what the models are fitted on.
- The validation set is used to tune model hyperparameters (such as which epoch should be used)
- The test set is used to evaluate the models’ performance on data it has not seen before.

Image created by finalist.

Coping with Imbalanced Classes



- Classifiers have trouble learning from highly imbalanced datasets
- Adaptive Synthetic Sampling Method (ADASYN) was used to perform oversampling of minority classes
  - ADASYN utilizes interpolation to add synthetic data points between existing data groups
  - ADASYN will synthesize more data in areas where it is difficult to distinguish classes.
  - Oversampling was only applied on the training split.

Graph created by finalist.

Feature Importance: Top 10

Importance

7.27% - Destination port

5.47% - The total number of bytes sent in initial window in the forward direction

5.42% - Average segment size observed in the forward direction

4.91% - The average number of bytes in a sub flow in the forward direction

4.89% - Standard deviation length of a packet

4.20% - Variance length of a packet

4.08% - Mean size of packet in forward direction

4.00% - The total number of bytes sent in initial window in the forward direction

3.89% - Maximum size of packet in forward direction

3.75% - Mean length of a packet

Support Vector Machine

- SVMs find a hyperplane in N-dimensional space (where N equals the number of features) that distinctly classifies data points.
- SVMs maximize the margin between the hyperplane and class data points closest to the hyperplane.
- Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane.

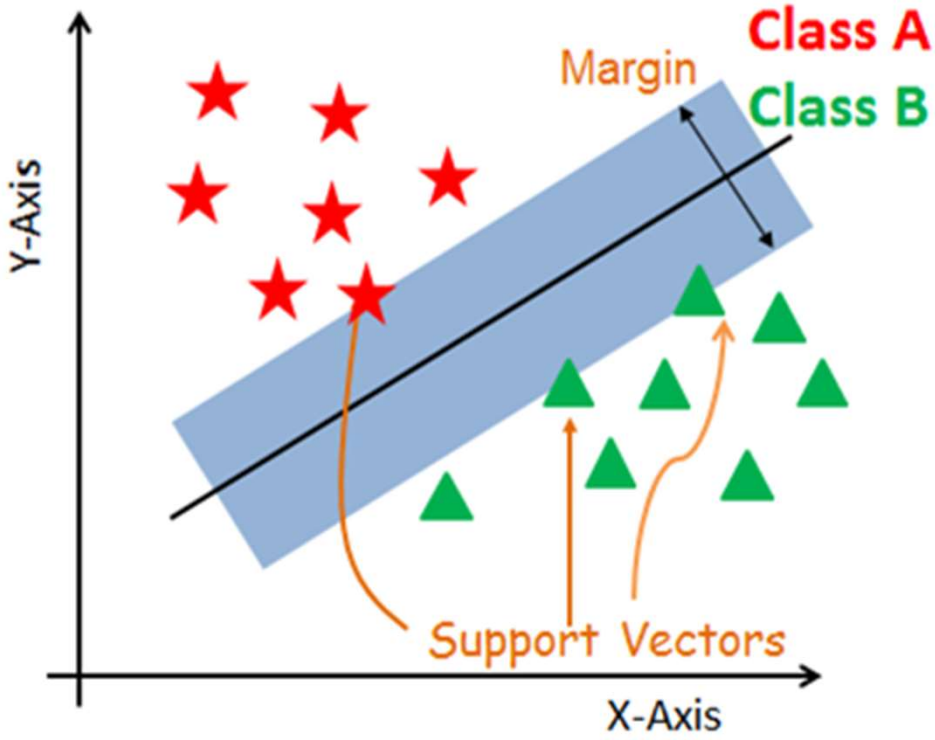


Image taken from DataCamp

K-Nearest Neighbors

- Data records processed by the KNN model are represented as geometric points.
- When making predictions, the model determines the k points that are closest to the unclassified point. The predicted class is the one that is the most common of the k-nearest points.
  - k is a hyperparameter that is determined before training

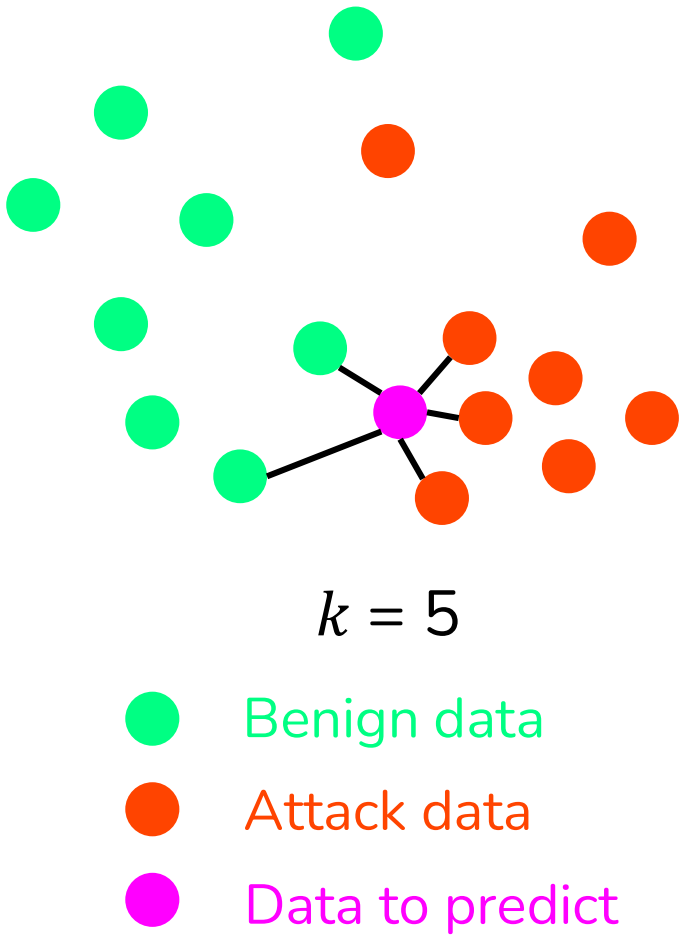
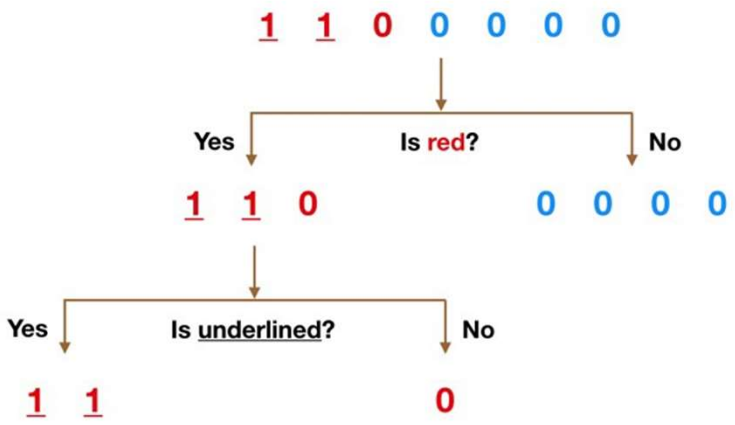


Image created by finalist.

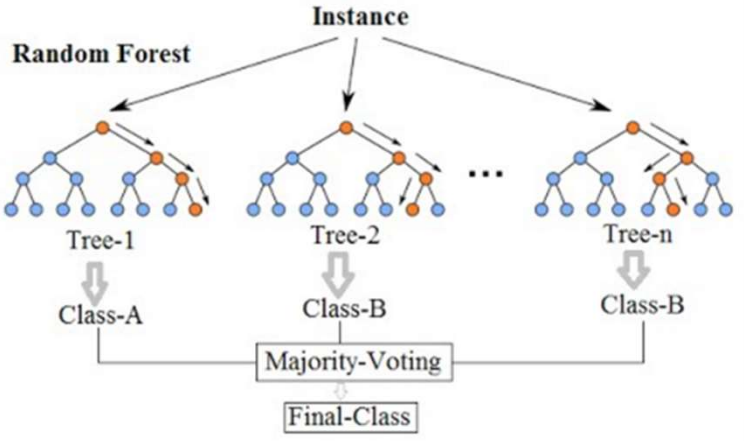
Decision Trees & Random Forests

- Decision Trees make predictions by splitting the data into subgroups based on the features until each subgroup corresponds to a single class.
- Random Forests utilize many decision trees to make predictions. The trees are trained using an ensemble method known as bagging (short for bootstrap aggregation). This means that each tree is trained on a random subset of the original data.
- The final prediction in a Random Forest is made by a majority vote of the trees.

Decision Tree



Random Forest



Images taken from TowardsDataScience and Medium.com.

Gradient Boosted Trees

- Like Random Forests, Gradient Boosted Trees is another ensembling technique that attempts to produce a precise classification model based on combining weaker models.
- In contrast to bagging, boosting improves models through a sequential process of increasing the weights of difficult to classify data points until they are correctly classified.
- Gradient Boosted Trees make a final prediction based on a weighted average of individual trees.

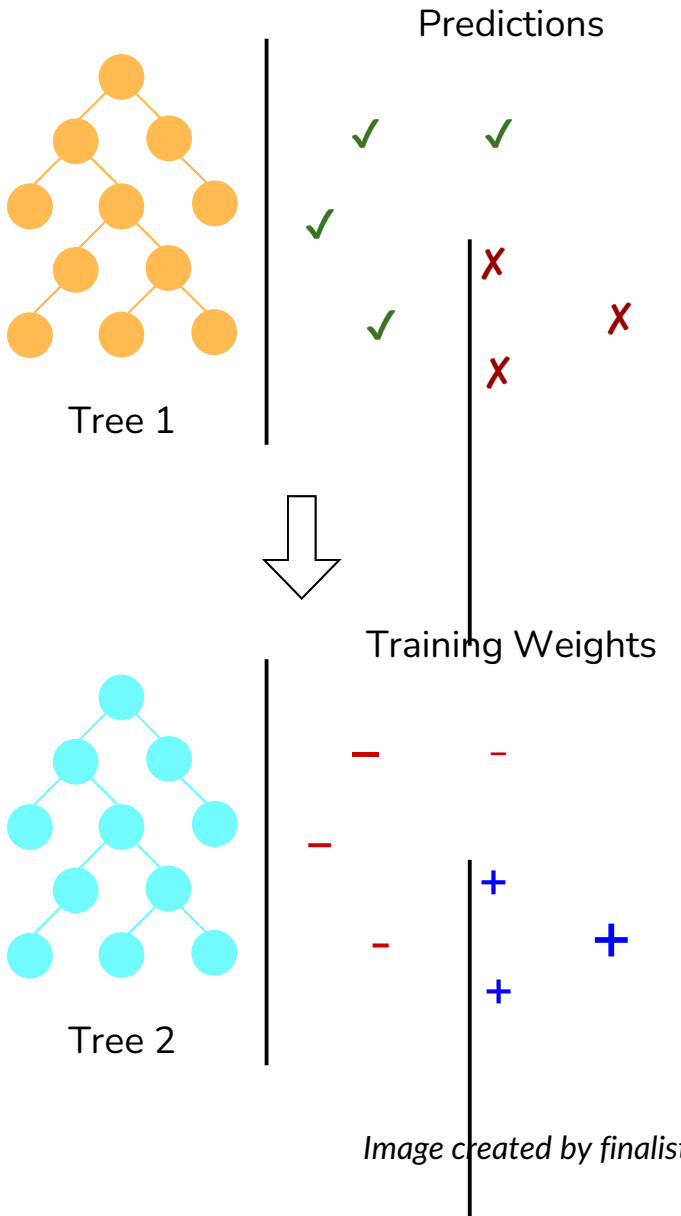


Image created by finalist.

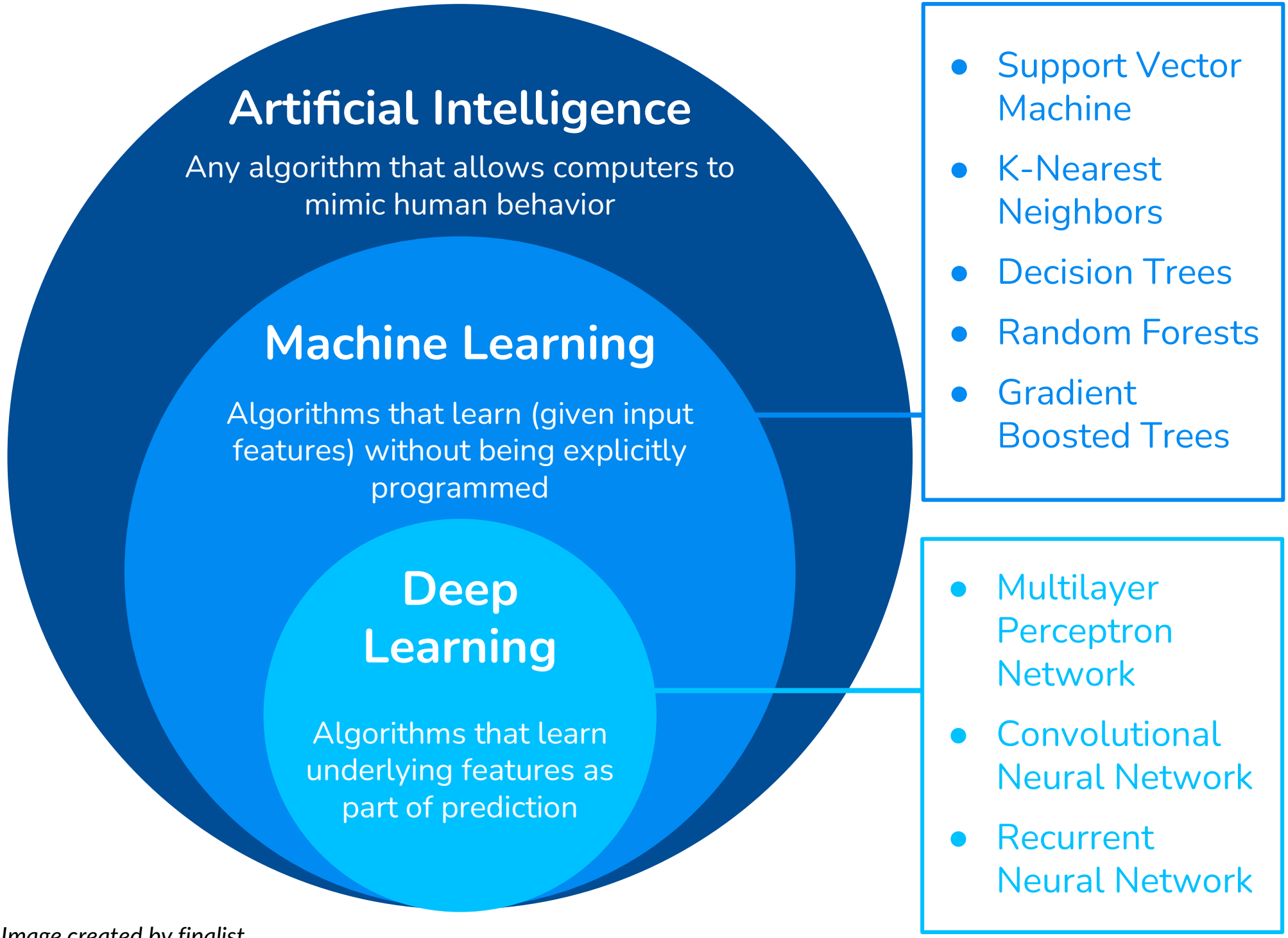
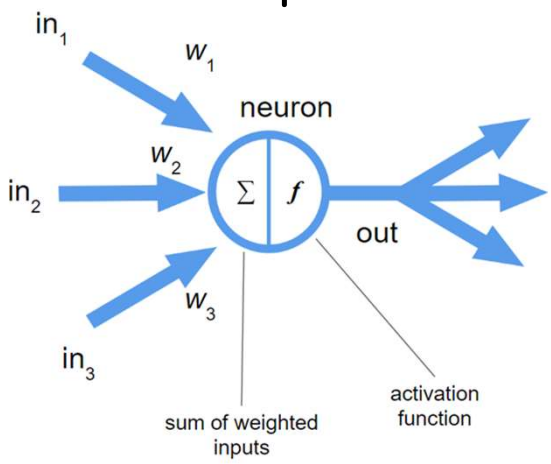


Image created by finalist.

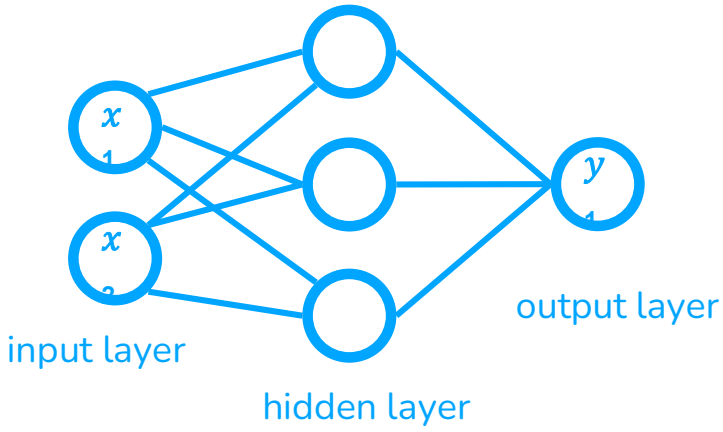
Multilayer Perceptron Networks

- Multilayer Perceptron Networks consist of many interconnected perceptrons.
- A Perceptron mimics biological neurons in that it receives input values from other neurons and, when a certain threshold is met, outputs a signal that determines what type of feature is perceived.
- The output layer of a Multilayer Perceptron Network creates a classification or prediction.

Perceptron



MLP Network



Images created by finalist.

Convolutional Neural Networks

- The “convolution” in a CNN is a function that extracts and condenses pixel data from the original image while maintaining the critical features of that image, including spatial relationships.
- Pooling layers that extract the maximum pixel values from the feature detector further condense the information and provide a scale invariant representation of the image that can be used to det

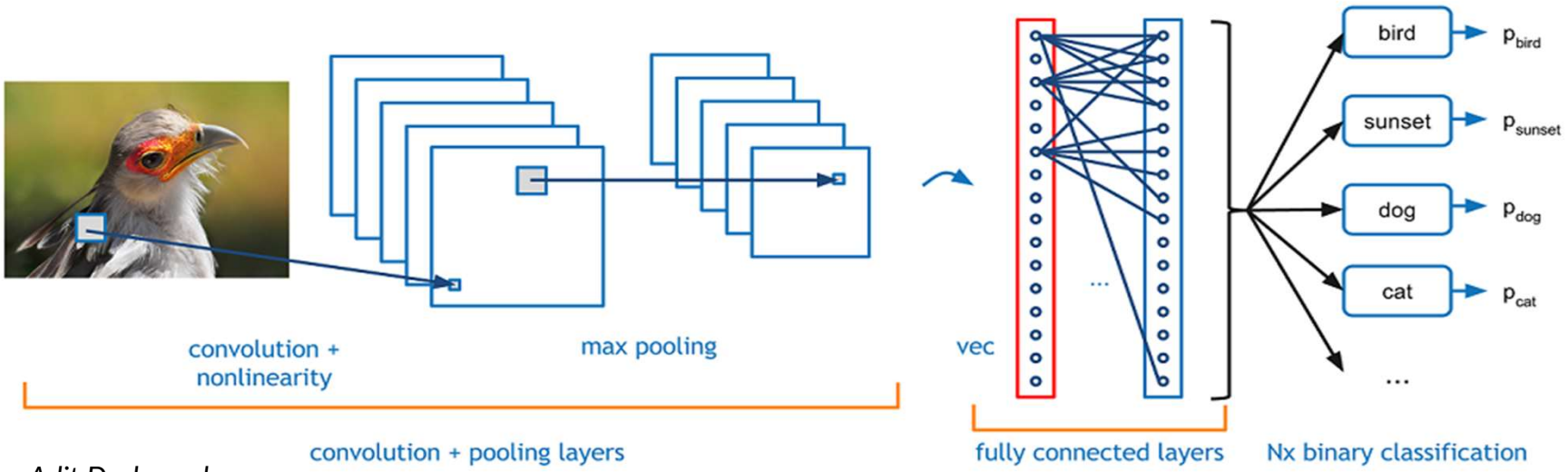


Image taken from Adit Deshpande.

Recurrent Neural Networks

- RNNs recognize sequential characteristics and patterns in data in order to make predictions.
- RNNs accomplish this by utilizing feedback loops to process data. This gives RNNs two sources of input: the data and the output of the neural network from previous time steps.
- The feedback loops allow information to persist, giving RNNs what is often described as memory.

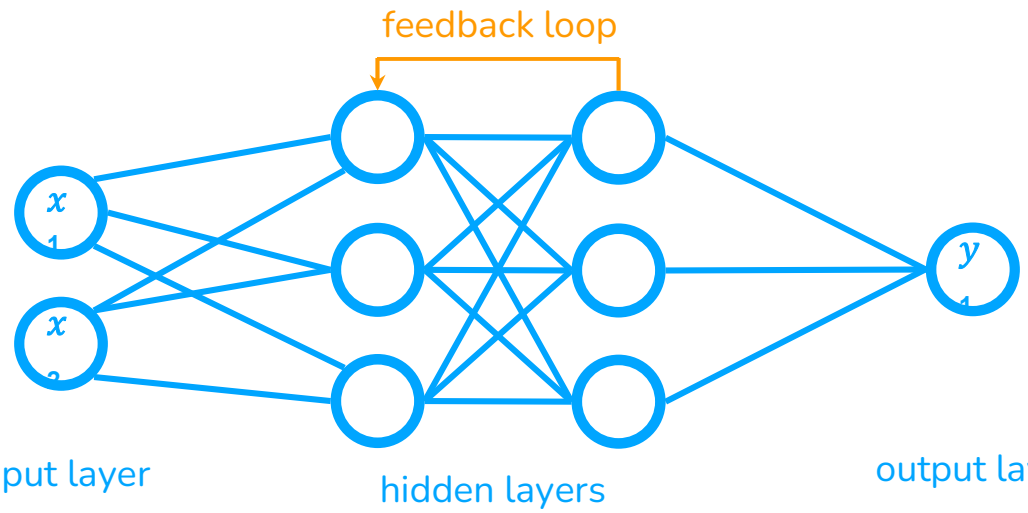


Image created by finalist.

Hypothesis

If different artificial intelligence models are compared for detecting network intrusions, then the Recurrent Neural Network will provide the best performance because network packets are sequential in nature and RNNs are designed to make accurate predictions given sequential data.

Materials

This project required a programming language installed on a personal computer that utilizes a graphics processing unit (GPU) and 16 gigabytes of random access memory. The GPU was used to train the neural network models. The dataset used for analysis was downloaded from an online source.

Procedures

- Download cyber intrusion dataset (see Dataset for details)
- Read in and preprocess the data (see Data Preprocessing for details)
- Train each model on the training split of the processed data
- For neural network models, choose the best epoch based on validation split scores
- Evaluate each model on the validation split of the data by generating predictions from each model and gathering the precision, recall, and F1 scores (see Evaluation Metrics for details)
- Repeat training and evaluation of each model, trying different hyperparameters to get the best performance from each model
- Evaluate the final version of each model on the test split of the data
- Compare the F1 scores for the best of each model
- Create tables and graphs to display results of the comparison

Variables

Independent: the artificial intelligence model used (SVM, KNN, DT, RF, GBT, MLP, CNN, or RNN)

Dependent: F1 scores (see Evaluation Metrics for details)

Constants: Dataset, data preprocessing, computer used, programming language used

Evaluation Metrics

Confusion Matrix

		Actual	
		Attack	**Benign**
Predicted	**Attack**	True Positives	False Positives
	Benign	False Negatives	True Negatives

Precision

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

Penalizes false positives

Recall

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

Penalizes false negatives

F1 Score

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Harmonic mean of Precision and Recall

Adjusted Hyperparameters

Support Vector Machine:

- Regularization constant (C)
- Gamma value

K-Nearest Neighbors:

- k value
- Distance or uniform weighted neighbors

Random Forests and GBTs:

- Number of trees
- Maximum depth of trees
- Number of features used per tree

Neural Networks:

- Number of hidden layers
- Number of neurons per layer
- Batch size
- Optimizer
- Learning rate
- Activation function
- Weight initialization
- Dropout
- Batch normalization

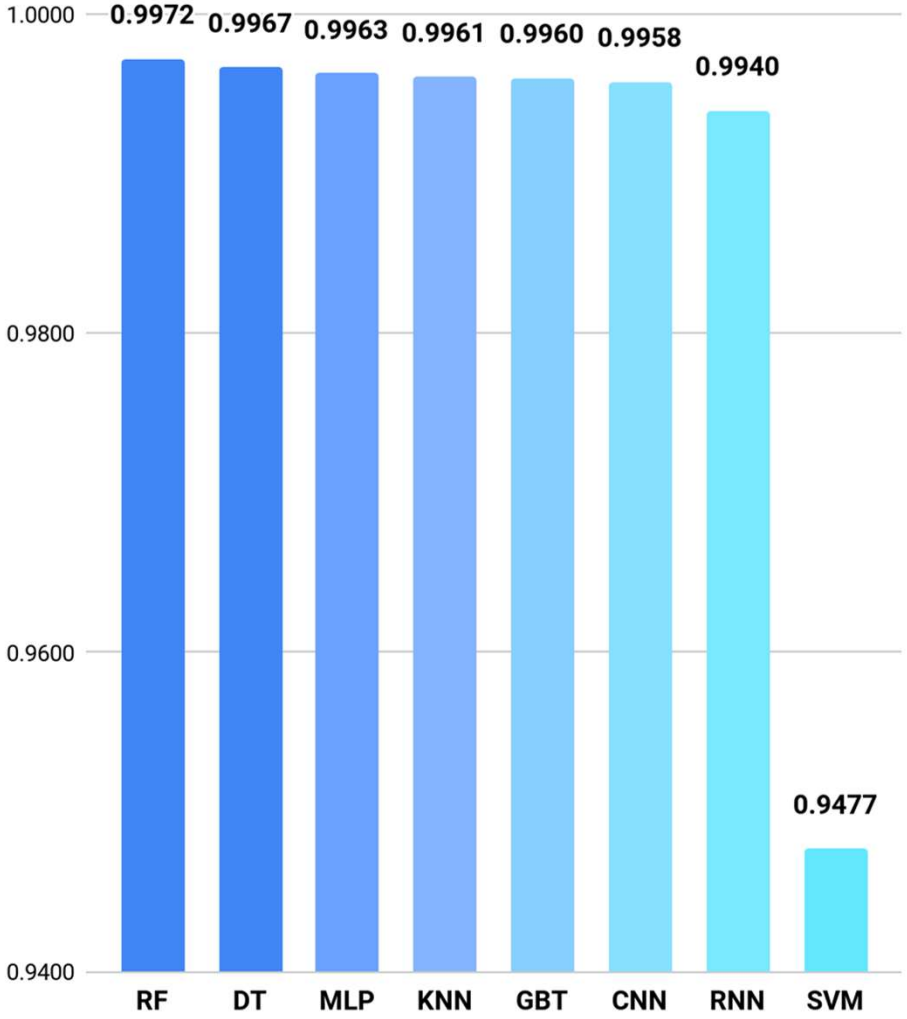
Data Table

Model	Binary			Multiclass	
	Precision	Recall	F1 Score	F1 Micro	F1 Macro
Random Forest	0.99553	0.99888	0.99720	0.99880	0.95220
Decision Tree	0.99573	0.99767	0.99670	0.99866	0.94290
Multilayer Perceptron Network	0.99281	0.99978	0.99629	0.99769	0.82847
K-Nearest-Neighbors	0.99306	0.99909	0.99607	0.99878	0.87094
Gradient Boosted Trees	0.99217	0.99987	0.99601	0.99863	0.93333
Convolutional Neural Network	0.99199	0.99959	0.99577	0.99758	0.86214
Recurrent Neural Network	0.98879	0.99920	0.99397	0.99749	0.79941
Support Vector Machine	0.90092	0.99963	0.94771	n/a	n/a

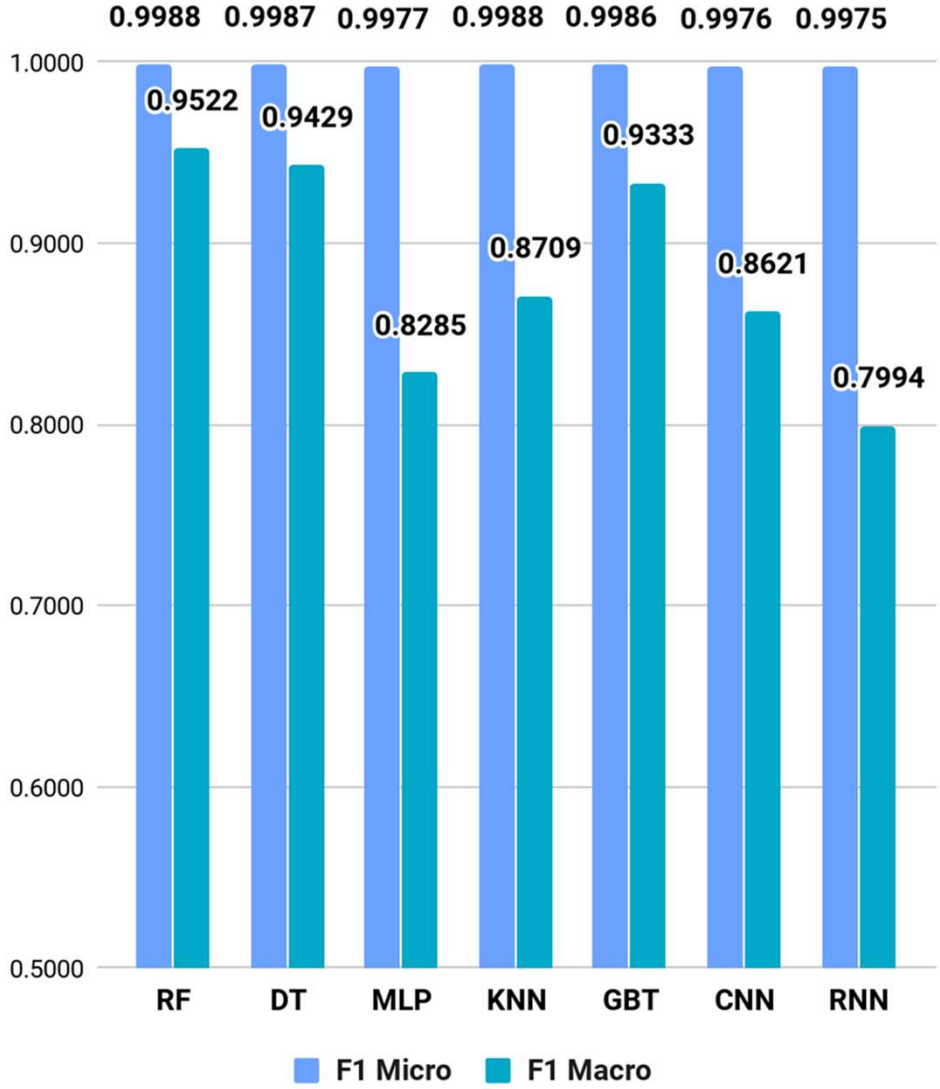
Table created by finalist.

Data Graphs

Binary F1 Scores



Multiclass F1 Scores



Graphs created by finalist.

Data Analysis

The models were tested with both a binary classification task (benign or malicious traffic) and a multiclass identification task (individual attacks need to be distinguished). For comparing models' performance on the multiclass task, the F1 macro-averaged score was used for comparison because it better captures how well all classes are predicted (compared to F1 micro).

- The **Random Forest** model was the strongest performing model overall because it scored the highest for both binary and multiclass tasks.
- The **individual Decision Tree** was the second best model overall because it scored the next best for both binary and multiclass.
- The third best model for the binary task was the **Multilayer Perceptron Network**.
- The third best score for the multiclass task was from the **Gradient Boosted Trees** model.

Conclusion

My experiment found that the Random Forest was the most effective artificial intelligence model for detecting and identifying cyber attacks. The hypothesis that a Recurrent Neural Network would have the best performance was invalidated because the F1 scores for the Recurrent Neural Network were lower than most of the other models tested in this project. The results instead show that simpler models such as K Nearest Neighbors and Decision Trees were better at detecting cyber attacks overall. The Random Forest, which is an ensemble method of combining Decision Trees, was the winner in distinguishing benign from malicious attacks as well as discriminating between different types of cyber attacks.

Implications and Next Steps

The outcome of this project demonstrates the viability of using artificial intelligence to detect cyber attacks with great accuracy and a low false alarm rate. The development and deployment of artificial intelligence enhanced network intrusion detection systems show great promise for minimizing critical personnel gaps and reducing the frequency of destructive and costly security breaches.

Further research on AI NIDS models can be carried out by:

- Utilizing transfer learning (weights that have been pre-trained on other cyber attack datasets)
- Ensembling different types of models together
- Use different ensembling techniques (e.g. stacking)
- Investigating unsupervised learning methods (such as clustering)

References

Accenture Security. (2019, March 6). The cost of cybercrime. Retrieved from https://www.accenture.com/us-en/insights/security/cost-cybercrime-study

Ackerman, R. (2019, January 27). Too few cybersecurity professionals is a gigantic problem for 2019. Retrieved from https://techcrunch.com/2019/01/27/too-few-cybersecurity-professionals-is-a-gigantic-problem-for-2019/

Federal Bureau of Investigation. (2019, April 22). IC3 annual report released: Report shows cyber-enabled crimes and costs rose in 2018. Retrieved from https://www.fbi.gov/news/stories/ic3-releases-2018-internet-crime-report-042219

Ford, M. (2018). *Architects of intelligence: The truth about AI from the people building it.* Birmingham, UK: Packt Publishing

Karn, U. (2016, August 11). An intuitive explanation of convolutional neural networks. https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

Marr, B. (2018, September 24). What are artificial neural networks – A simple explanation for absolutely anyone. Retrieved from https://www. forbes.com/sites/bernardmarr/2018/09/24/what-are-artificial-neural-networks-a-simple-explanation-for-absolutely-anyone/#479ca8031245

Olah, C. (2015, August 27). Understanding LSTM Networks. Retrieved from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Paul, F. (2019, January 14). Top 10 IoT vulnerabilities. Retrieved from https://www.networkworld.com/article/3332032/top-10-iot-vulnerabilities.html

Shinder, D. (2005, July 13). SolutionBase: Understanding how an intrusion detection system (IDS) works. Retrieved from https://www.techrepublic.com/ article/solutionbase-understanding-how-an-intrusion-detection-system-ids-works/