Deep Learning based Dynamic Analysis for Malware Detection

By

M S Mithran

Strategy and Synergy for Security

Society for Electronic Transactions and Security

Chennai

10th Feb 2022

Malware Detection

- Malware is an acronym for **mal**icious soft**ware**
- Script or binary code that performs some malicious activity
- Malware can come in different formats
  - Executables
  - Binary shell code
  - Script
  - Firmware

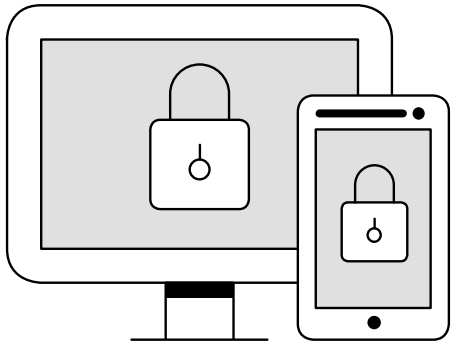
Malware Types

RANSOMWARE

Encrypts the contents on infected systems and demands payment in cryptocurrencies majorly in bitcoins

TROJAN

Designed to damage, disrupt, steal, or in general inflict some other harmful action on your data or network



SPYWARE

Gather info about a person or organization and sends information to another entity in a way that is harmful to the user

WORMS

Malicious code that replicate itself without attaching itself to a software program in order to cause damage.

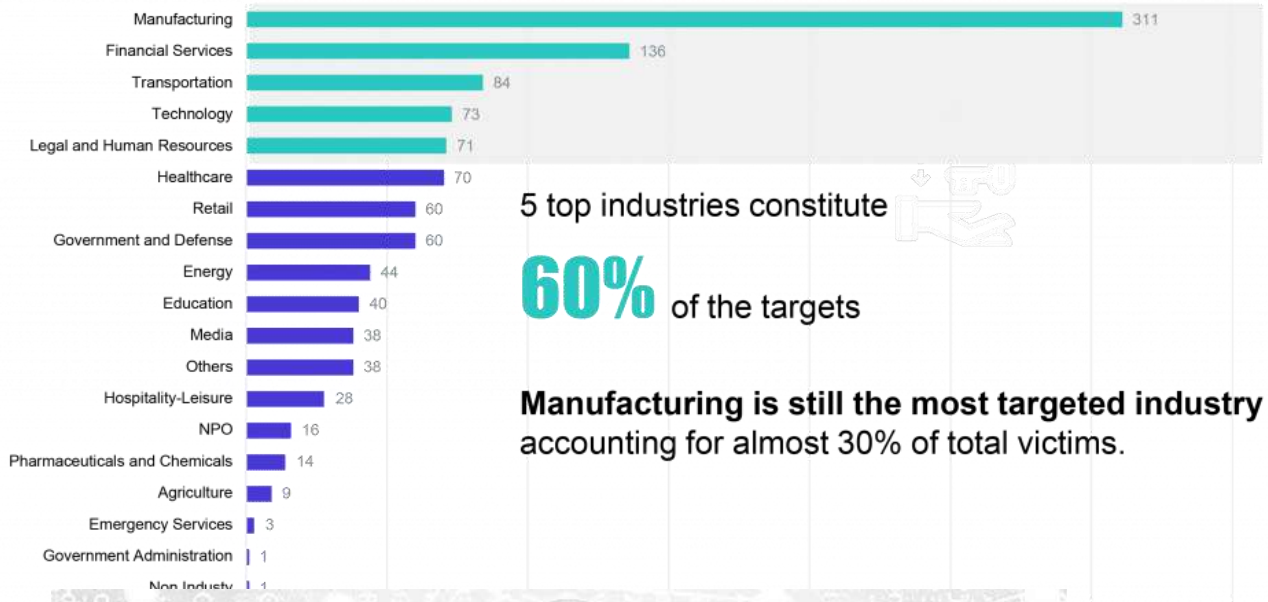
ADWARE

Presents unwanted advertisements to the user of a computer in the form of a pop-up or in an "unclosable window"

VIRUS

Alters the way a computer operates and is designed to spread from one computer to another. Requires host file to attach itself

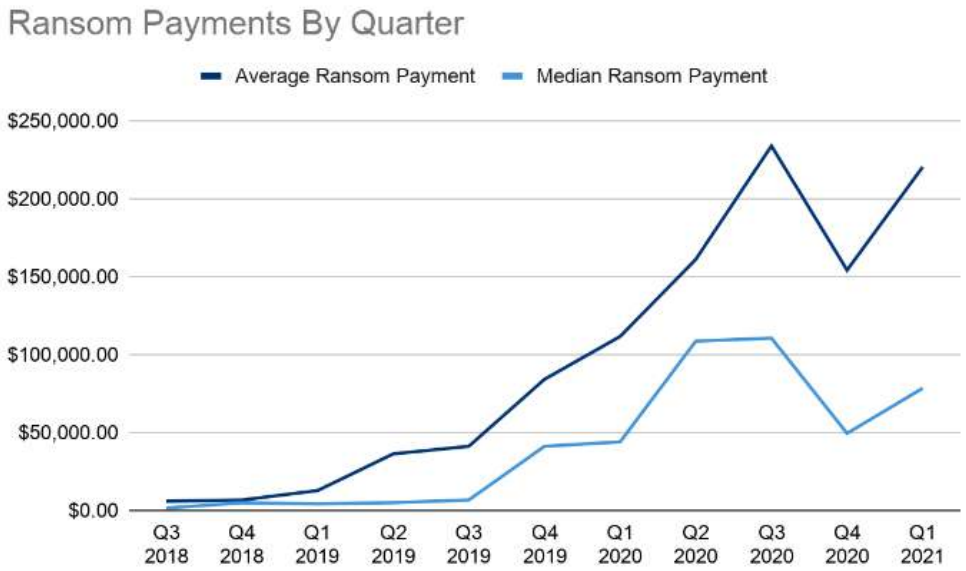
Impact

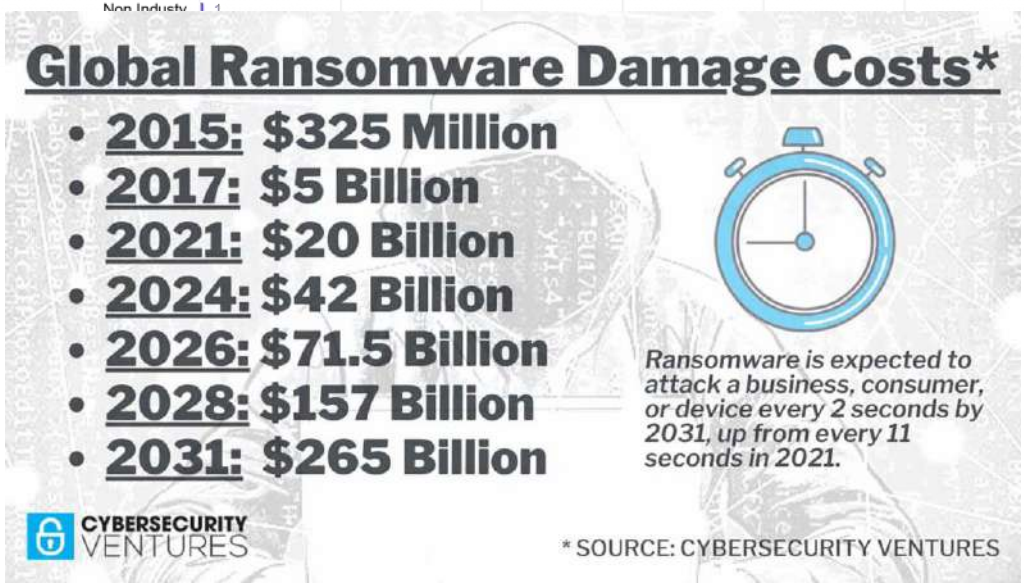


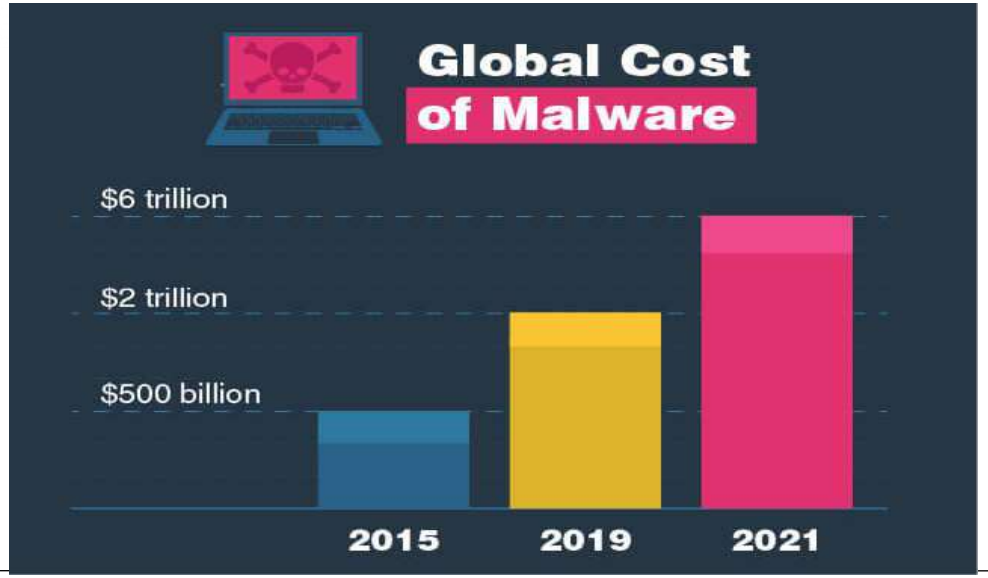
5 top industries constitute

60% of the targets

Manufacturing is still the most targeted industry accounting for almost 30% of total victims.







Malware Analysis

Analyzing the behavior, functionality, and impact of malware samples on systems

STATIC ANALYSIS

Without executing the malicious code
Tools: PEiD, Yara, IDA Pro etc

DYNAMIC ANALYSIS

While executing malicious code
Tools: Cuckoo, Detours etc

MEMORY ANALYSIS

After executing malicious Code
Tools: Volatility, PIN, Valgrind etc

Static Analysis

- Analysis of source code without executing the application
- Portable Executable file sections
  - A PE file contains a header and some more important sections. Under these sections there is some useful information.
    - .text: This contains the executable code.
    - .rdata: This sections holds read only globally accessible data.
    - .data: Stores global data accessed through the program.
    - .rsrc: This sections stores resources needed by the executable.

Dynamic Analysis

- Dynamic analysis is the process of testing and evaluating a program – while software is running.
- Also referred as dynamic code scanning, dynamic analysis improves the diagnosis and correction of bugs, memory issues, and crashes of an application during its execution.
- Dynamic analysis is extracting the API calls made by a binary file while in execution.
- Tools used for Dynamic Analysis
  - Cuckoo Sandbox
  - Detours etc.
- Using one of the Dynamic analysis tool like cuckoo is used to run and analyse malware files and generate analysis result of the behaviour of malware while in execution.
- The log file contains API calls made during execution, registry modifications and the information such as heap memory address and process address

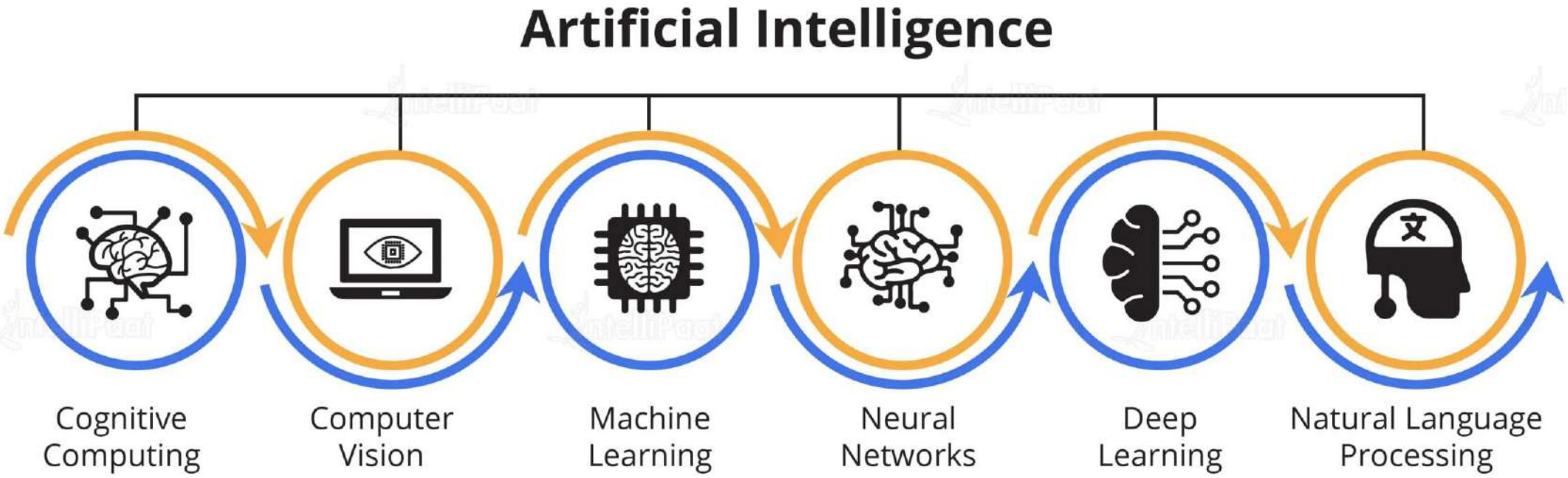
Dynamic Analysis

- APIs are provided by the operating system to access the low level hardware through system calls for the application programs
- The attackers use the same set of API to do malicious activities.
- Features:
  - Similarity in the API call sequence between files in the same class must be greater than the similarity between the files in the different classes
  - N-gram based method to analyse the call sequence called API-call-grams

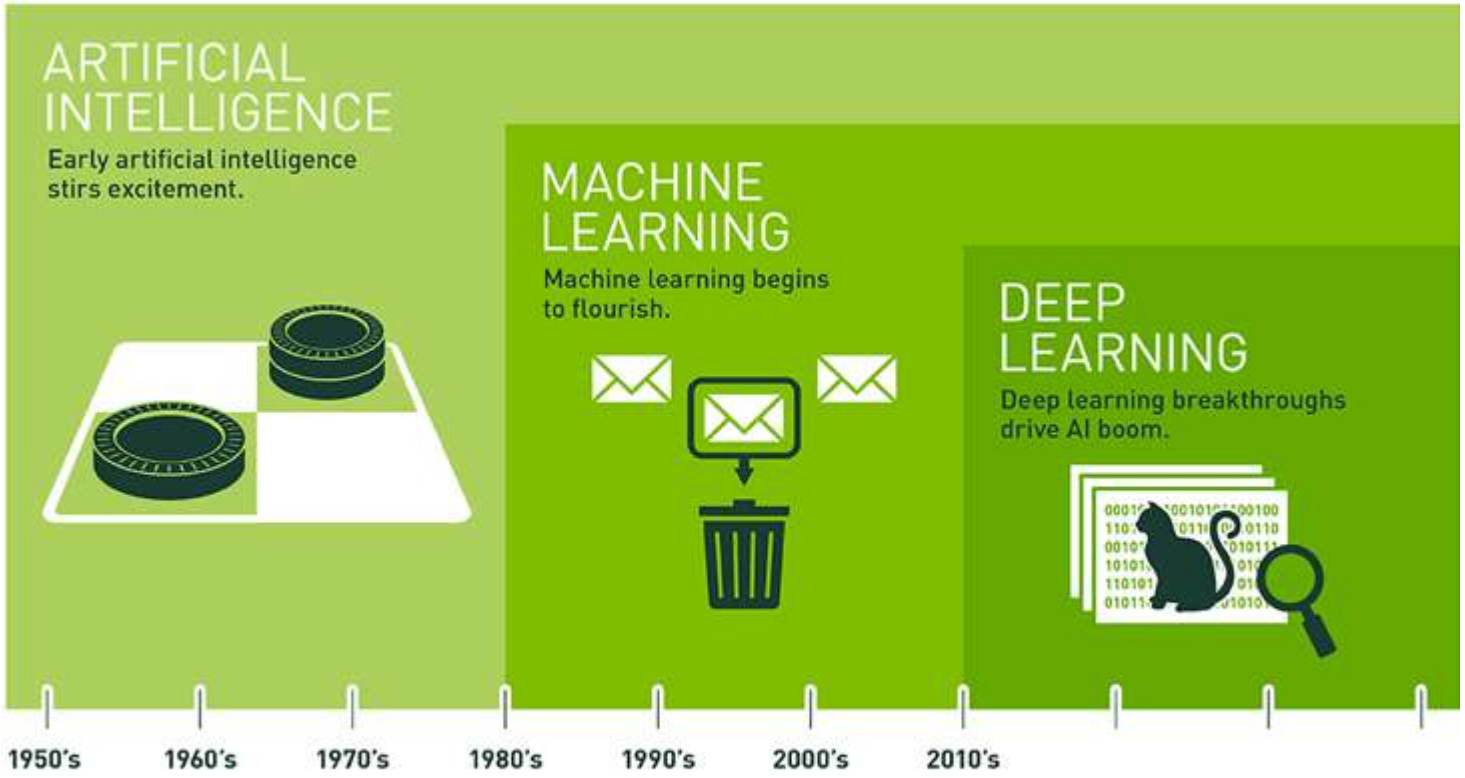
Introduction

Artificial Intelligence

- It is the ability of a machine to perform cognitive functions as humans do, such as perceiving, learning, reasoning and solving problems.
- The benchmark for AI is the human level concerning in teams of reasoning, speech, and vision.



Why AI is Hype now?



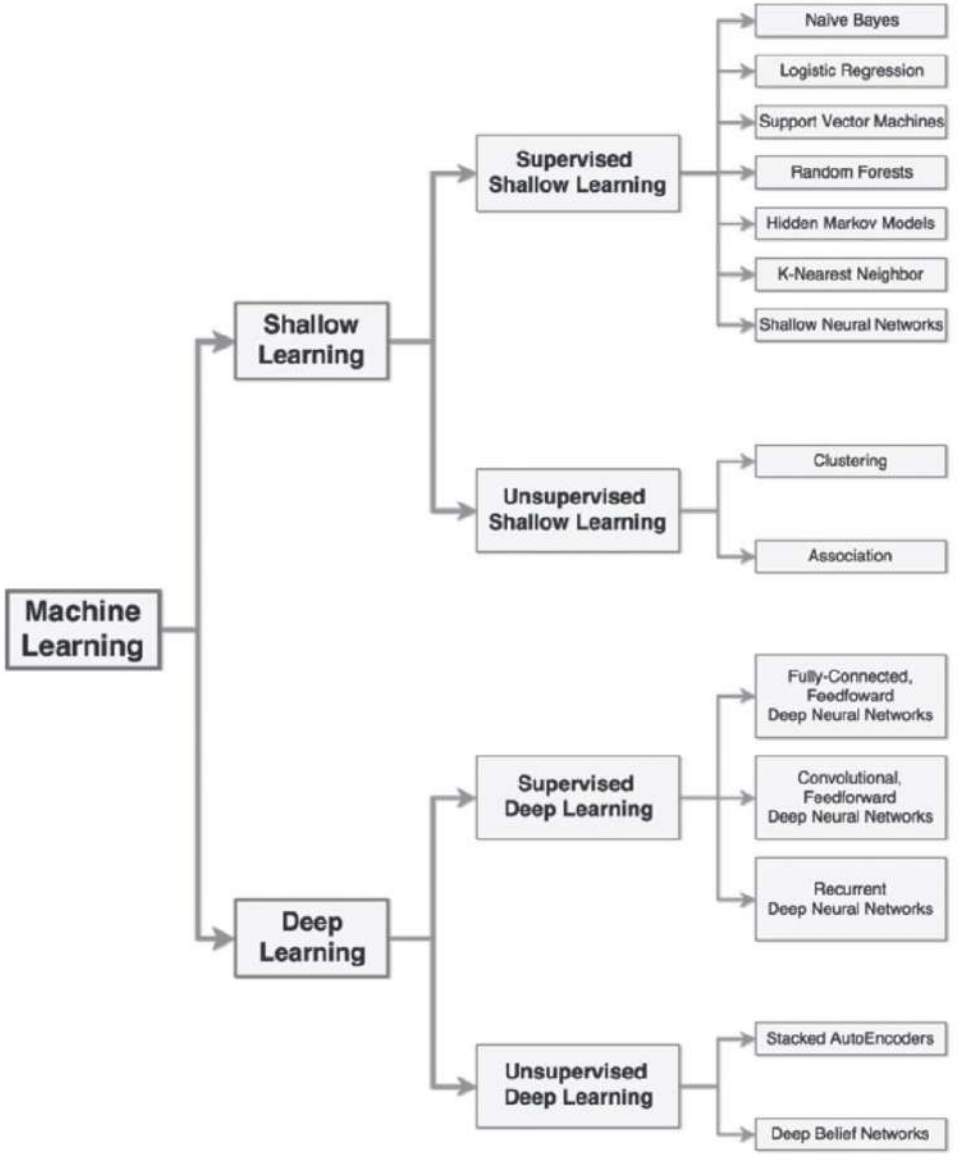
Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

Availability

- Hardware
- Data
- Algorithm

Machine Learning Steps

Machine Learning overview



Introduction

Deep Learning

- Part of Machine Learning field of learning representation of data
- Exceptionally effective at learning patterns
- Tries to minimize the difference between its prediction and expected output
- By doing so, it tries to learn the association/pattern between given inputs and outputs

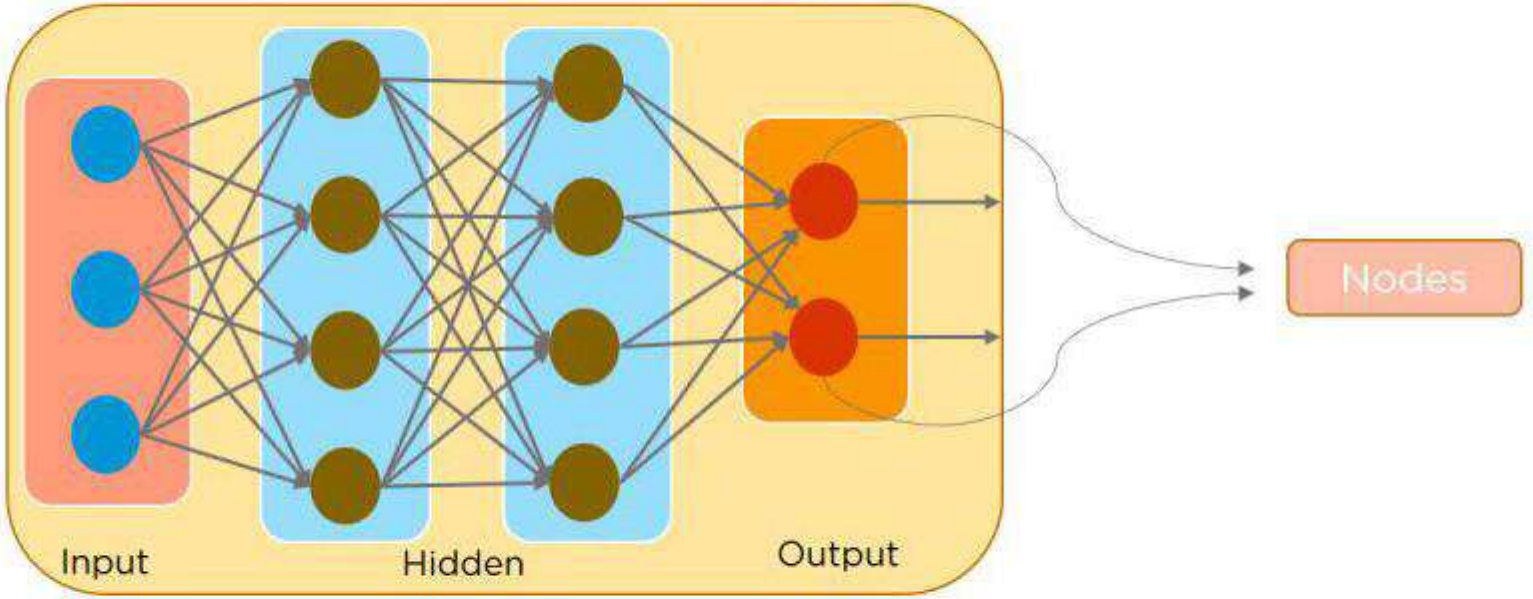
Types of Deep Learning

1. Convolutional Neural Networks (CNNs)
2. Long Short Term Memory Networks (LSTMs)
3. Recurrent Neural Networks (RNNs)
4. Generative Adversarial Networks (GANs)
5. Radial Basis Function Networks (RBFNs)
6. Multilayer Perceptrons (MLPs)
7. Self Organizing Maps (SOMs)
8. Deep Belief Networks (DBNs)
9. Restricted Boltzmann Machines(RBMs)
10. Autoencoders

Deep Learning

Human brain inspired systems which replicate the way humans learn

Consists of 3 layers of network (input, hidden and output)

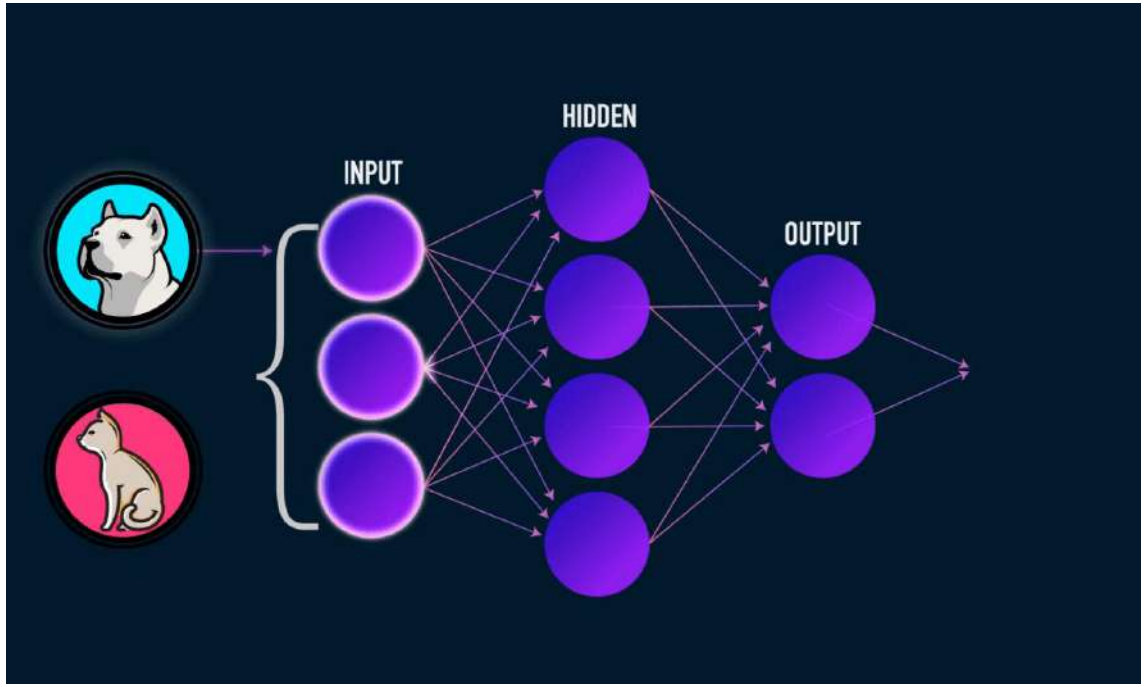


Used in Deep Learning algorithms like CNN, RNN, GAN, etc

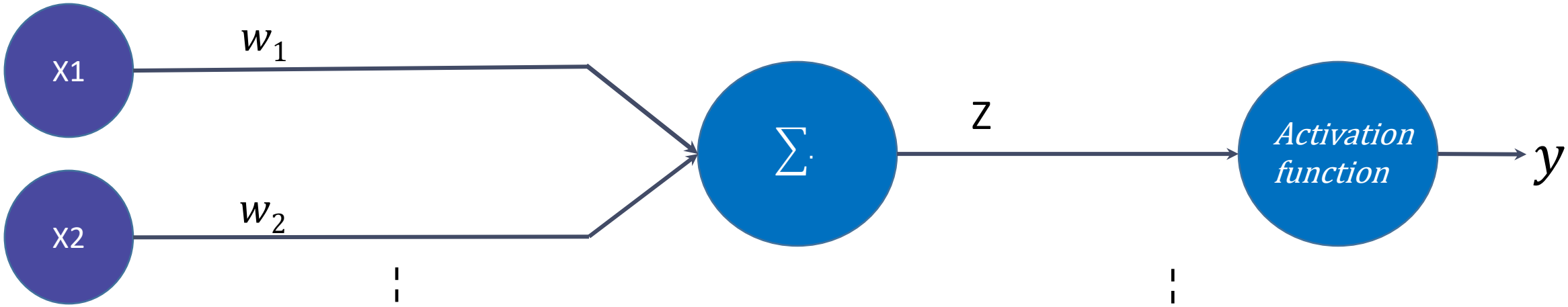
Each layer contains neurons called as nodes that perform various operations

Deep learning

- Uses artificial neural networks to perform sophisticated computations on large amounts of data
- Neural network is a structured layer of artificial neurons which are also called as nodes
- The structure of these artificial neurons and their network results in different forms of neural networks
- Each form of neural network has its own applications



Deep learning - Perceptron



$$x = [x_1, x_2, \dots\dots, xn]$$

$$w = [w_1, w_2, \dots\dots, wn]$$

$$\sum = (x_1 \times w_1) + (x_2 \times w_2) + \cdots\cdots + (xn \times wn)$$

$$\sum = x.w$$

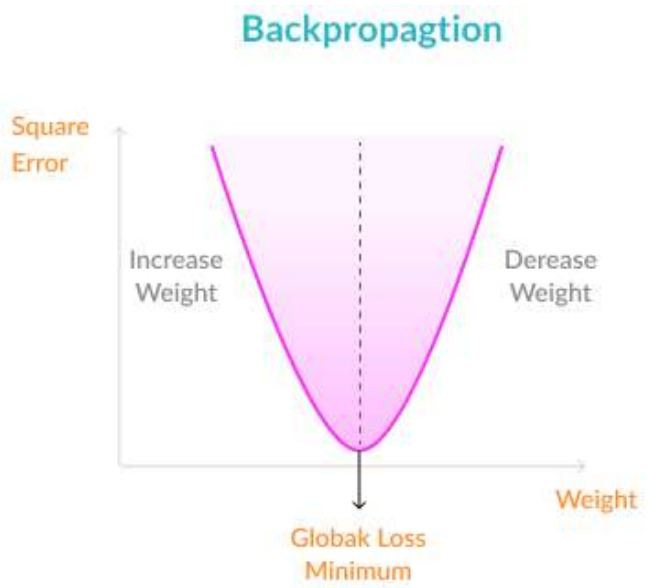
$$z = x.w + b$$

$$y = \sigma(z) = \frac{1}{1 + e^{-z}}$$

Activation Function

- Function that is added into an artificial neural network in order to help the network learn complex patterns in the data.
- It takes in the output signal from the previous cell and converts it into some form that can be taken as input to the next cell.
- The most important feature in an activation function is its ability to add non-linearity into a neural network.
- Different types of Activation Functions are:
  - Sigmoid
  - Softmax
  - Tanh
  - ReLU (Rectified Linear Unit)

Backpropagation



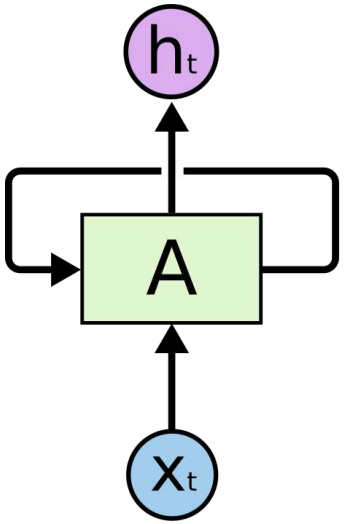
- Computes the gradient of the loss function with respect to the weights
- Objective of backpropagation is to change the weights for the neurons, in order to bring the error function to a minimum.
- Two type of loss functions are used namely:
  - Cross-entropy
    - Used for error calculation between estimated and predicted probability distribution
  - Mean squared error
    - Used for error calculation between estimated and predicted quantity for regression problems

Recurrent Neural Network

- Output from previous step are fed as input to the current step
- In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words
- An RNN remembers each and every information through time. It is useful in time series prediction only because of the feature to remember previous inputs as well. This is called Long Short Term Memory.
- Recurrent neural network are even used with convolutional layers to extend the effective pixel neighborhood.

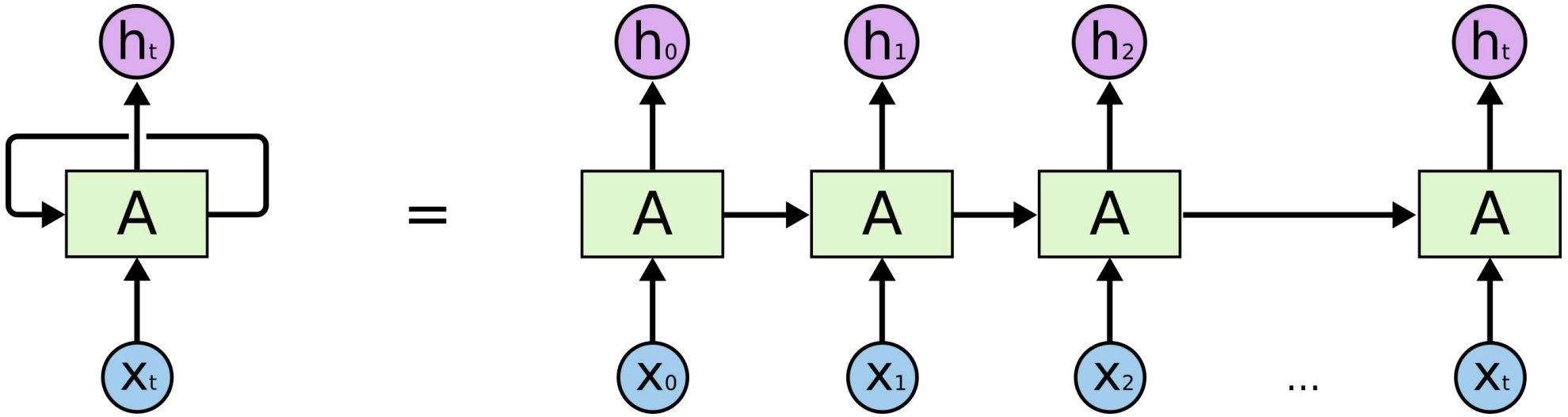
Recurrent Neural Network

- Gradient vanishing and exploding problems.
- Training an RNN is a very difficult task.
- It cannot process very long sequences if using tanh or relu as an activation function.



Long Short Term Memory

- In order to overcome the issue of vanishing gradient problem in RNN, LSTM are used
- Uses gating mechanism that controls the memorizing process
- Information in LSTMs can be stored, written, or read via gates that open and close



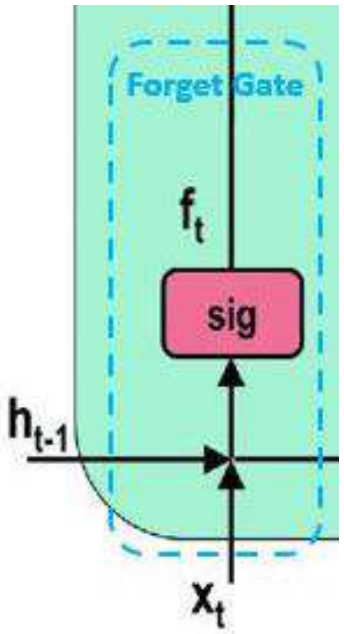
Long Short Term Memory

LSTM Cell

- Consists of:
  - Forget Gate
  - Input Gate
  - Cell State
  - Output Gate
- Forget Gate
  - Responsible for deciding what information is to be thrown away or kept from the last step
- Input gate
- Cell State
  - Cell state serves as the memory of an LSTM. This is where they perform way better than vanilla RNN's when dealing with longer sequences of input.
- Output Gate
  - The cell state obtained from above is passed through a hyperbolic function called tanh so that the cell state values are filtered between -1 and 1

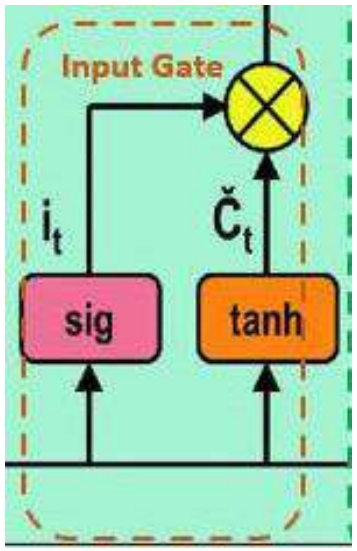
Long Short Term Memory

Forget Gate



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

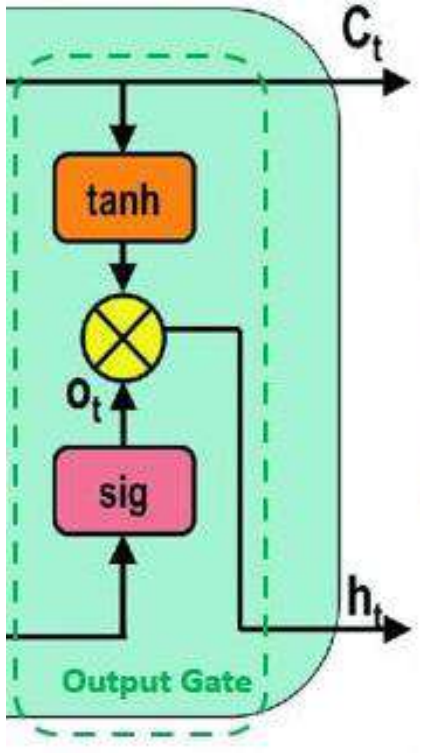
Input Gate



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

$$C_t = tanh(W_c[h_{t-1}, x_t] + b_c)$$

Output Gate

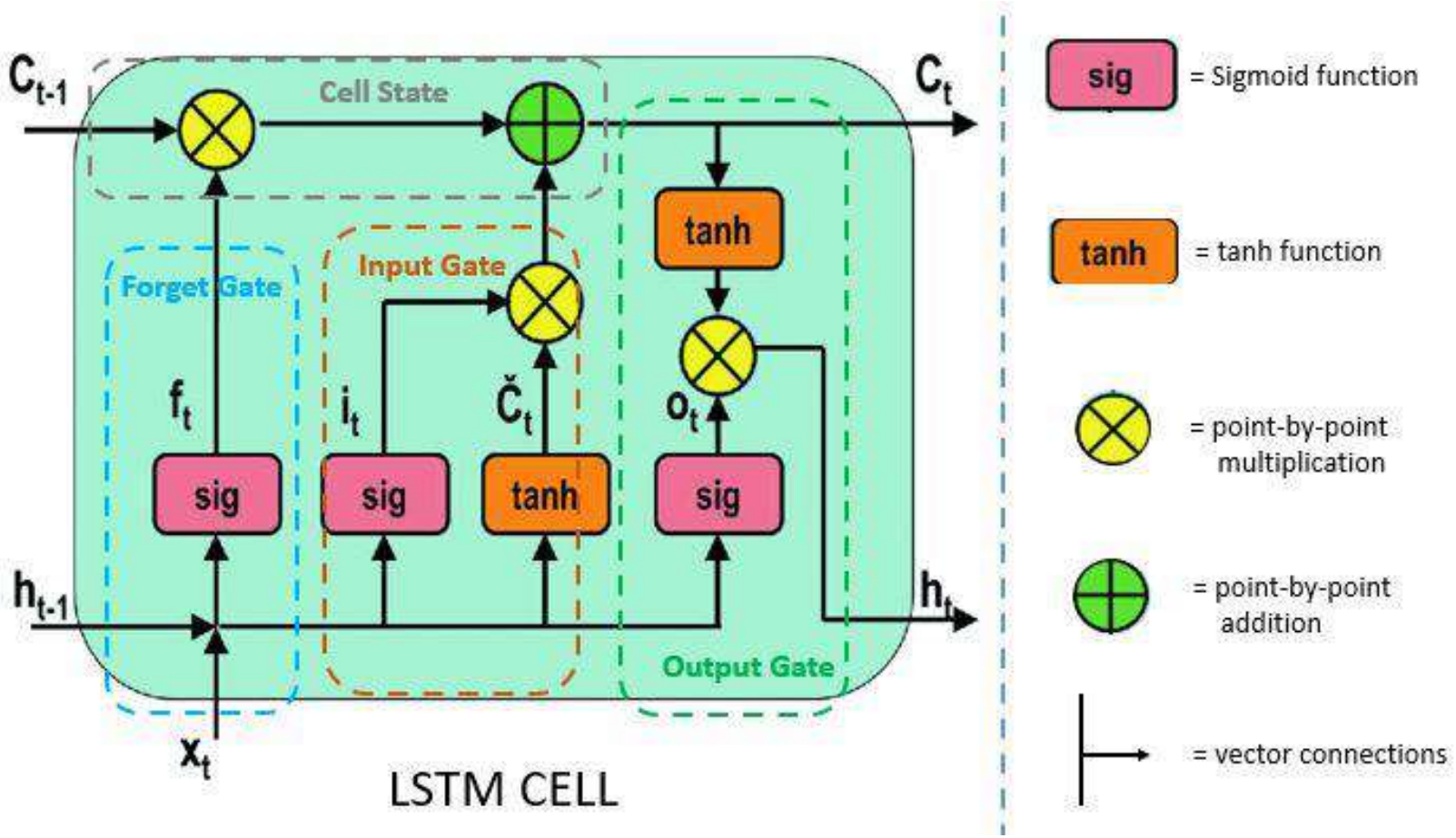


$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \cdot \tanh(c_t)$$

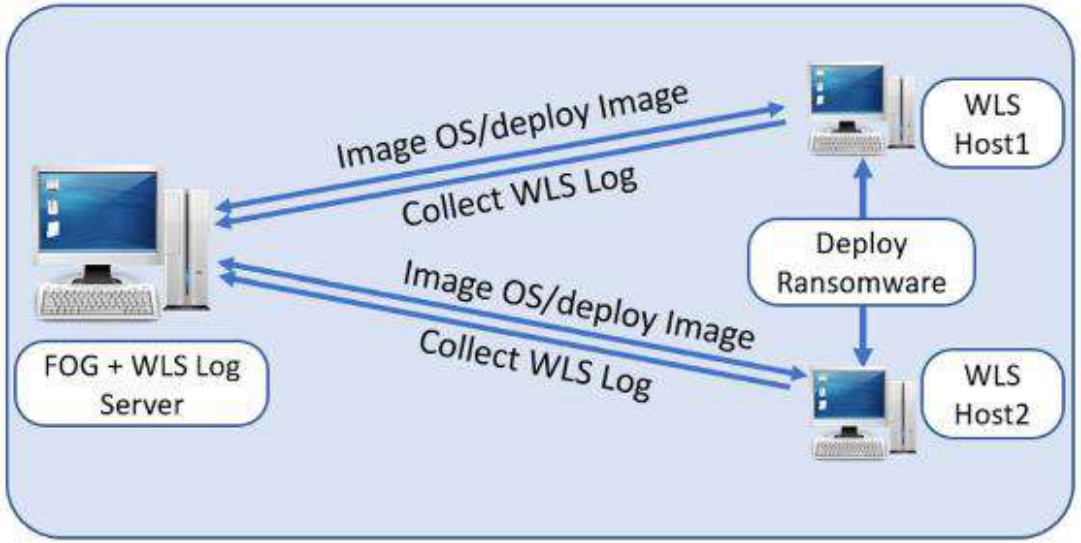
Long Short Term Memory

LSTM Cell

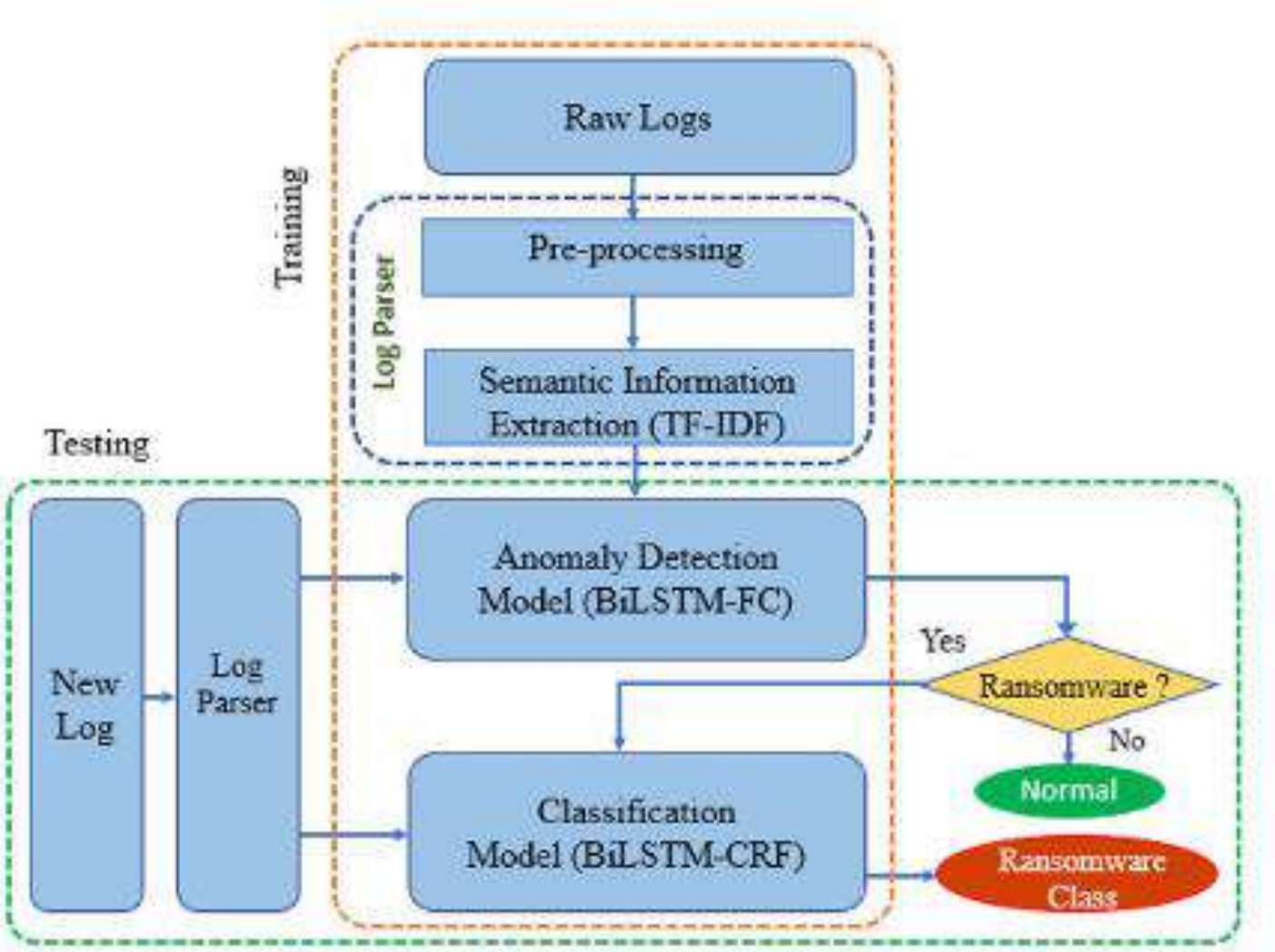


Testbed

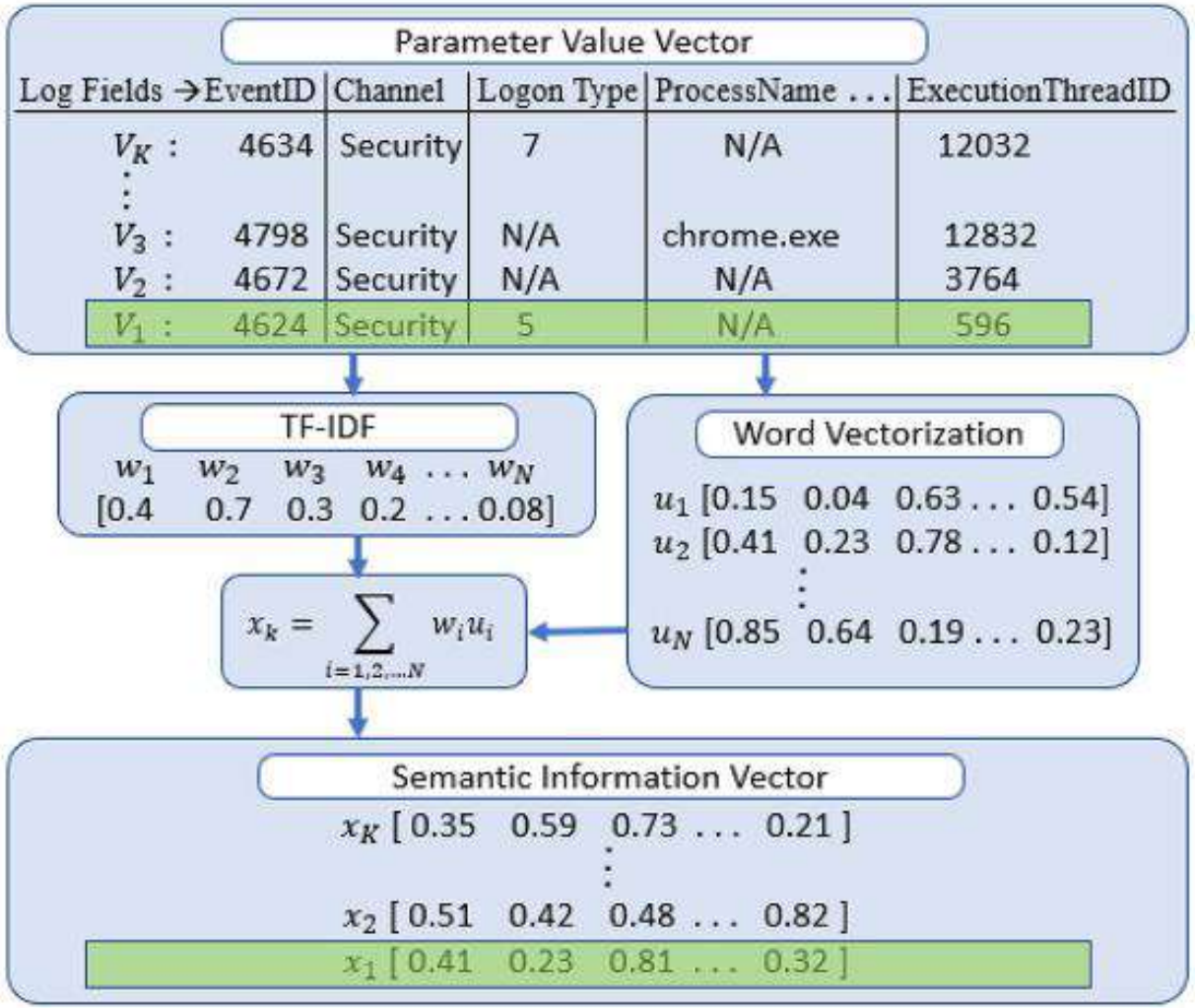
- Testbed is needed for running the malware files in isolated environment
- A simple testbed consists of three physical/bare-metal machines: one Linux server and two Windows 7 clients, Host1 and Host2.
- Dynamic analysis tool like, FOG server or cuckoo server is deployed on the FOG server
- Log is collected by running the malware on the Host systems



Framework

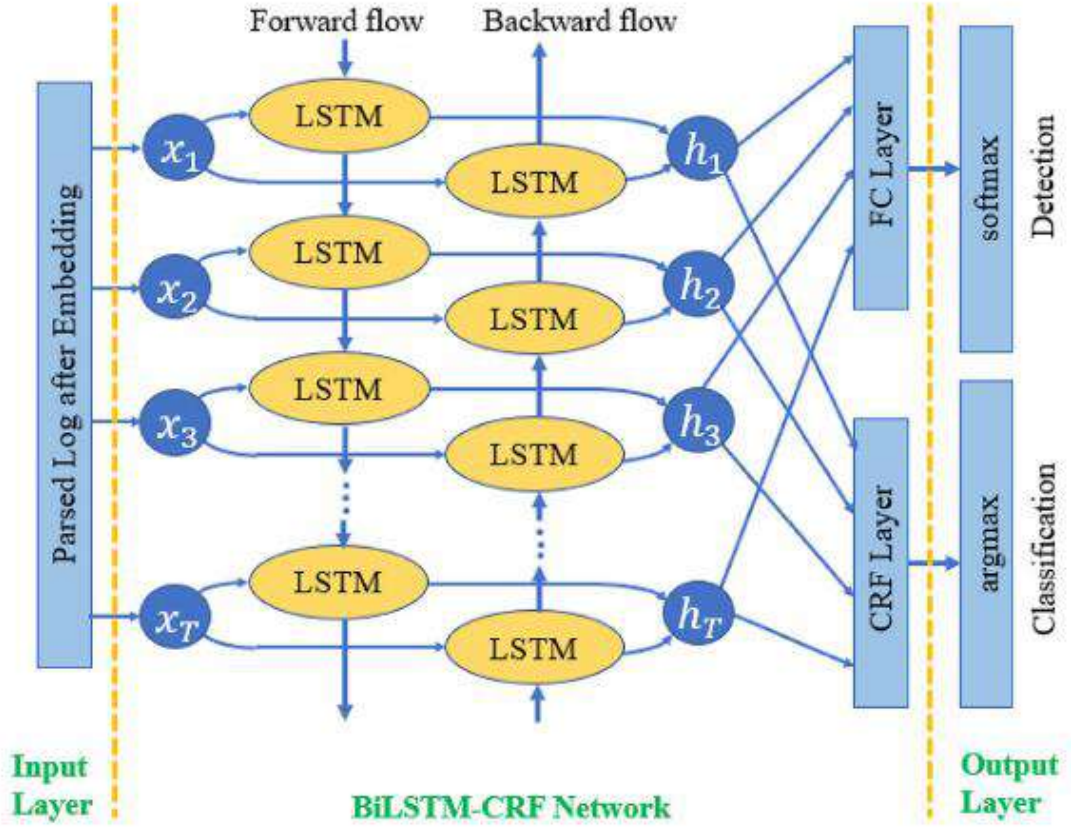

Training
Raw Logs
Log Parser
Pre-processing
Semantic Information Extraction (TF-IDF)
Testing
New Log
Log Parser
Anomaly Detection Model (BiLSTM-FC)
Yes
Ransomware ?
No
Normal
Classification Model (BiLSTM-CRF)
Ransomware Class

Log Pre-processing



Neural Network Model

- Bidirectional Long Short Term Memory – Conditional Random Field
- BiLSTM is a variant of LSTM that connects two hidden layers of opposite directions to the same output.



Future work

- Using more advanced neural networks like seq-to-seq models, autoencoders can increase the accuracy of the overall model.

Thank You