**Jagadeesh Bavineni**

**40219221**

**1. Genetic algorithm overview:**

A Genetic Algorithm (GA) is a programming method which uses evolution as a model to solve problems. A population of potential solutions is optimised in the direction of a predetermined fitness according to the Darwinian theory of evolution and survival of the fittest [1].

Using a data format resembling a chromosome, GA employs natural selection to evolve the chromosomes through mutation, recombination, and selection. The technique typically starts with a population of chromosomes that are created at random to represent all potential solutions to a problem [1]. [2]The term "population," which is made up of "individuals," refers to the set of solutions. Every individual has a "chromosome," which is made up of "genes" (positions) and "alleles" (values). A tuple of integers called a chromosome serves to specifically identify an individual. This is typically a tuple of binary numbers, however real numbers are also possible. "Generations" is the name given to evolved populations. A fitness function is an evaluation function to quantify the quality of each solution, this evolutionary strategy is used to evolve this set by merging the best solutions and producing new set of solutions, iterating until a stop criteria have been met. When two individuals cross over, their chromosomes are divided, and for each gene, a random allele from one parent is chosen for the binary case or the mean of the parents' alleles is used for the real number genes. This process results in a new chromosome that is used to refer to the new individual, also known as the"offspring."Mutation is a process that, according to a certain probability, randomly modifies some members of the population's genes. Three factors will have a significant impact on the effectiveness of the algorithm as well as the applications when we use GA to solve various challenges[3]. They are the following: (1) fitness function; (2) individual representation; and (3) GA parameters. Applications and/or implementation are frequently taken into consideration when determining these factors.

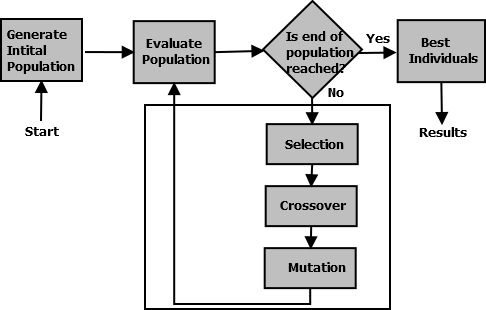


Figure 1.1- Execution flow of Genetic algorithm [5]

**2. Methodology:**

Sazzadul, Abdul et al.[1]. Proposed a system of algorithm for GA basing their work on KDD 99 dataset, that can be divided in two phases, the phases of precalculation and detection. Major steps in the precalculation phase, in which a set of chromosomes is constructed using training data. In the following phase, this set of chromosomes will be used for comparison. The detection phase, where a population is generated for test data and evaluated using various techniques (selection, crossover, mutation), leading to the prediction of the test data's type. In this stage, the precalculated set of chromosomes is utilised to ascertain  each population's chromosome's fitness.

**2.1. Fitness function**

The created new chromosomes, which stand in for possible solutions, were shown and put up against the practise data. To find any attack patterns that might have been saved in the database, this was done. To determine the fitness value, all attributes were assessed using the training data set. The goal of employing the fitness function is to choose the most fit individuals to move on to the next phase and produce the subsequent batch of chromosomes. In their article[4], Hamizan et al. established the formula below.



Where "a" denotes the number of attacks discovered through population and data set comparison, "A" denotes the total number of attacks in the dataset, and "b" is the number of normal connections discovered out of the dataset's total number of normal connections, B. If the fitness value is near to 1, the chromosome is regarded as being of good quality.

**2.2. Crossover and mutation**

Hamizan et al.[4] selected 10 fit individuals based on the fitness score. Each chromosome was cloned five times, resulting in a total of 50 parents for the chromosomes. An example of this process is in Figure 2.

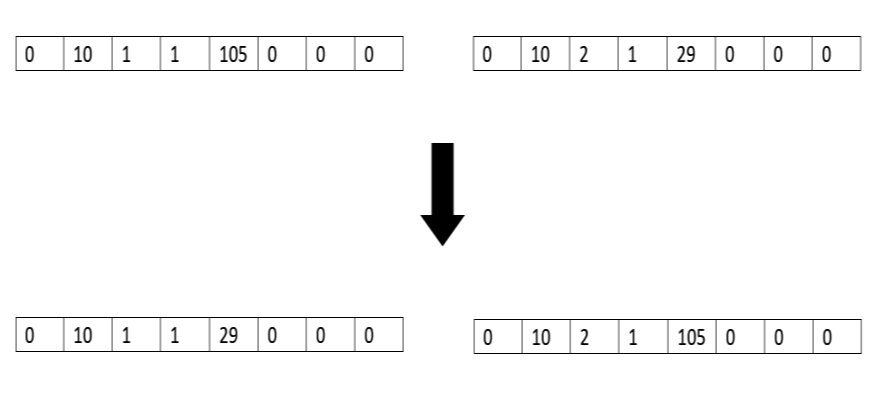


Figure 2.1 – Process of crossover [4]

Based on the specified crossover rate, crossover takes place according to the above Figure, this will determine how many traits from each parent will be passed down to both offspring. It is essential to ensure that the offspring maintain the acceptable range of values in each field. 25 of the 50 cloned chromosomes were chosen as "parent 1," while the other 25 were chosen as "parent 2." 50 new chromosomes (offspring) were created as a result of the crossover. The gene(s) on the new chromosome(s) were then slightly altered by mutating these chromosomes in accordance with the probability of mutation. The single mutation strategy was used to carry out the mutation process.

The procedure continues until there is just one generation left (one). The projected type is the set of chromosomes that the test data's sole surviving chromosome is most closely related to.

**3. EXPERIMENTAL RESULTS AND ANALYSIS**

According to the suggested methodology, the final outcomes depend on the variety of the data and how it is processed. Interactions between the testing dataset and the chromosomal set obtained after training were made throughout the testing procedure.

[1]The ratio of intrusions that were correctly identified to all incursions is known as the detection rate (DR), that is:



False positive rate (FP) is determined by dividing the total number of normal connections by the number of normal connections that are falsely classified as intrusions, that is



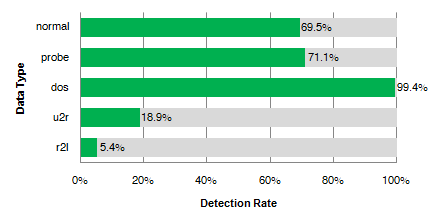


Figure- 3.1 Detection rate of each class[1]

Except for the normal data type, which was a result of disregarding nonnumerical properties, the system of Sazzadul et al.[1] operated effectively for the majority of classes. They had a higher success rate of denial of service and user-to-root detection, and an almost same rate of probe and remote-to-local detection than a highly benchmarked KDD 99 system. [5]When used in conjunction with other classifiers to optimise the classification process's parameters and the feature selection in intrusion detection systems, GA has been found to perform effectively. They fall short in a few areas, such as the fact that there is no guarantee that a genetic algorithm would discover a global optimum. Furthermore, the evolutionary algorithm has a complex representation of a problem space. They require the evolution of numerous fitness functions.

**References:**

[1] Sazzadul Hoque, M., Mukit, M. A., and Abu Naser Bikas, M., “An Implementation of Intrusion Detection System Using Genetic Algorithm”, arXiv e-prints, 2012.

[2] Angelo, Paulo & Drummond, André. (2018). Adaptive anomaly‐based intrusion detection system using genetic algorithm and profiling. Security and Privacy. 1. 10.1002/spy2.36.

[3] Ren Hui Gong, M. Zulkernine and P. Abolmaesumi, "A software implementation of a genetic algorithm based approach to network intrusion detection," Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Network, 2005, pp. 246-253, doi: 10.1109/SNPD-SAWN.2005.9.

[4] Suhaimi, Hamizan & Suliman, Saiful & Musirin, Professor Dr. Ismail & Harun, Afdallyna & Mohamad, Roslina. (2019). Network intrusion detection system by using genetic algorithm. Indonesian Journal of Electrical Engineering and Computer Science. 16. 1593. 10.11591/ijeecs.v16.i3.pp1593-1599.

[5] P. Mishra, V. Varadharajan, U. Tupakula and E. S. Pilli, "A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection," in IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 686-728, Firstquarter 2019, doi: 10.1109/COMST.2018.2847722.