Temasek Defence Systems Institute

Temasek Defence Systems Institute

NEURAL DETECTION OF MALICIOUS NETWORK ACTIVITIES USING A NEW DIRECT PARSING AND FEATURE EXTRACTION TECHNIQUE

Low Cheng Hong

Phillip Pace & Monique P. Fargues

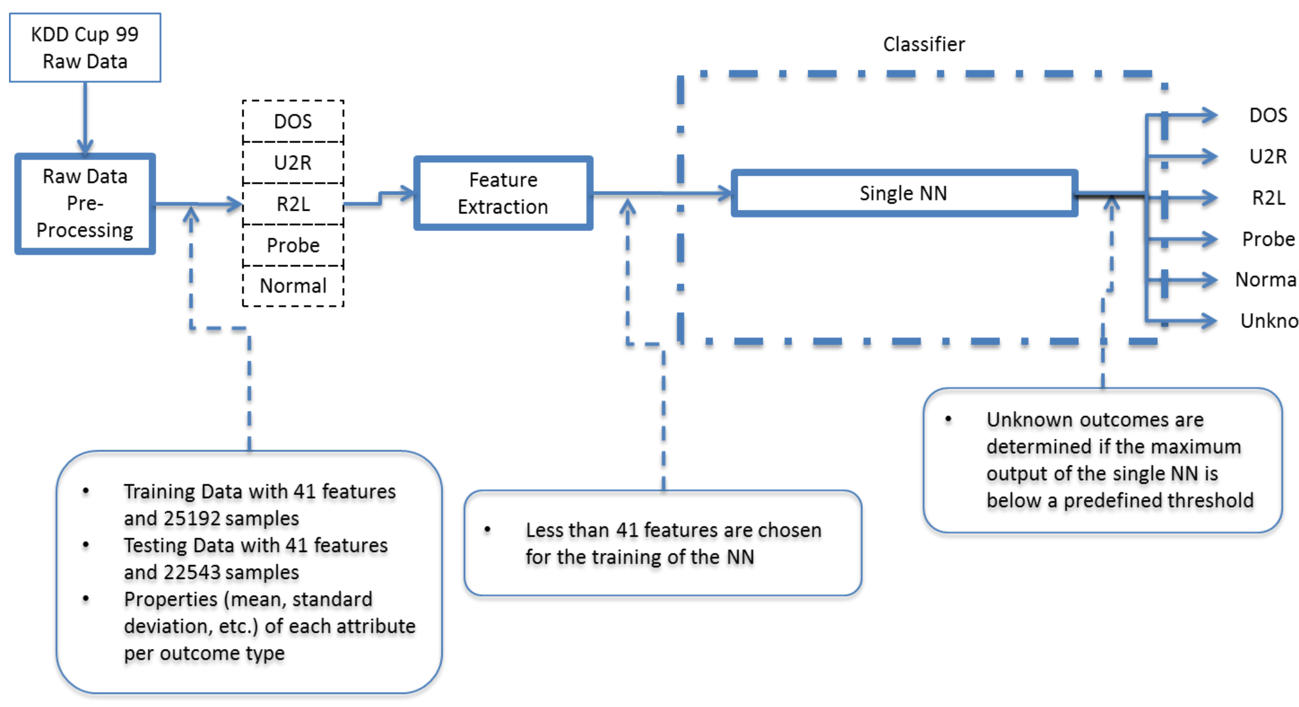
Objectives of thesis

- The growing threat of malicious network activities and intrusion attempts makes intrusion detection systems (IDS) a necessity. Network traffic includes attacker packets, normal packets, and victim packets; thus, IDS must work in a dynamic environment and requires continuous tuning to react against the evolving new attacks that exploit newly discovered security weaknesses.
- An IDS software prototype was developed with the following design goals:

1. Handle known classes of attacks;
2. Handle unknown classes of attacks and classify accordingly;
3. Operate in near real-time to classify continuous network traffic;
4. Retrain based on new network data with minimal disruption to real-time operations.

Main Research Ideas

- The IDS implementation was developed using MATLAB 2014b and is divided into three software modules: raw data pre-processing, feature extraction and classification.
- The raw data pre-processing stage takes the KDD Cup 99 raw data and converts it into a matrix of representative numerical data for the feature-extraction and classification stage
- The feature extraction stage uses the properties of each feature per outcome type to evaluate the features that are useful at the training stage. The first filtering mechanism removes all features which remain relatively constant across different outcome types. The second filtering mechanism calculates the correlation coefficient matrix of the features, and pairs of features which have a correlation coefficient value larger than or equal to 0.97 are identified. For each pair of highly correlated features, only one of the features is selected for the feature vector and used at the classification stage
- Two different classification architectures were explored and compared in terms of accuracy and computing times. The first architecture utilized a single NN for the classification process as illustrated in Figure 1. The second architecture utilized three separate NNs for the classification process as illustrated in Figure 2. This second structure was inspired by [1], where the training data was randomly separated into an arbitrary number of subsets for the purpose of temporal analysis of network traffic data, and each subset was used to train an individual classifier separately.
- The unknown type was determined through the use of a threshold value; if the maximum output value of the Neural Network (NN) is below the empirical threshold value of 0.8, this means that the NN is not able to match its output with one of the known output types with high confidence. In this case, the NN output defaults to the unknown type

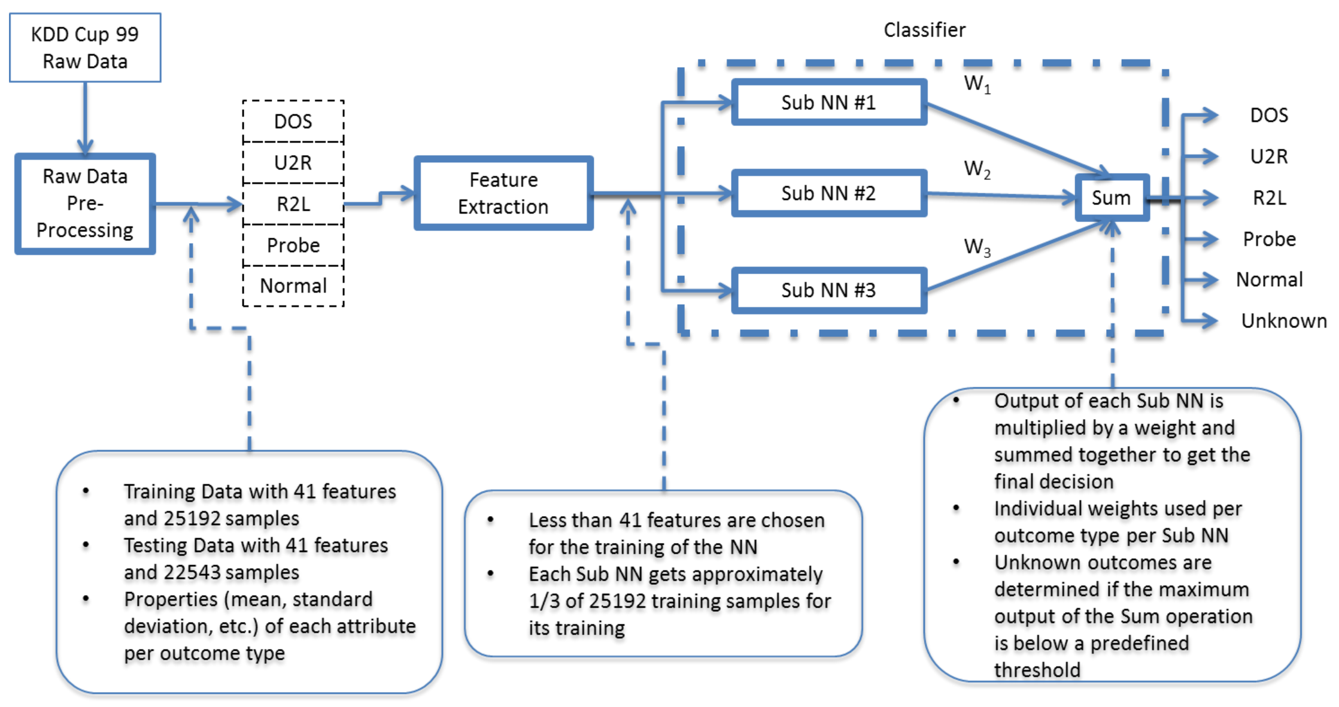


- Training Data with 41 features and 25192 samples
- Testing Data with 41 features and 22543 samples
- Properties (mean, standard deviation, etc.) of each attribute per outcome type

- Less than 41 features are chosen for the training of the NN

- Unknown outcomes are determined if the maximum output of the single NN is below a predefined threshold

Figure 1: Single NN-based IDS process flow



- Training Data with 41 features and 25192 samples
- Testing Data with 41 features and 22543 samples
- Properties (mean, standard deviation, etc.) of each attribute per outcome type

- Less than 41 features are chosen for the training of the NN
- Each Sub NN gets approximately 1/3 of 25192 training samples for its training

- Output of each Sub NN is multiplied by a weight and summed together to get the final decision
- Individual weights used per outcome type per Sub NN
- Unknown outcomes are determined if the maximum output of the Sum operation is below a predefined threshold

Figure 2: Three-parallel NN-based IDS process flow

Research Results

- Reducing the feature size does not degrade classification performances in either NN implementations; the feature-extraction stage successfully removed irrelevant features which did not serve to improve the NN classification capabilities.
- The overall training time is shorter for the three-parallel NN implementation and the three parallel NN implementation has comparable classification performance to the single NN implementation.
- The training time for the three-parallel NN implementation can be further improved, when three parallel computing threads are used to run the training of each sub NN.
- Classification results is on the order of 93% for denial-of-service attacks, 73% for probe and 96% for normal traffic. This is due to the large number of training samples for the DOS, Probe and Normal outcome types, which allows the NN to be sufficiently trained
- U2R and R2L outcome types have low probability of correct classification due to fewer training samples than DOS, Probe and Normal training samples. As a result, the training dataset is imbalanced, which may have led to insufficient training of the NN in classifying the U2R and R2L outcome types.

Benefits/Potential applications of the research

- Three-parallel NN implementation is comparable in classification performance to the single NN implementation, it was shown to be superior in terms of training time. This makes the three-parallel NN implementation a possible candidate for use in real-time applications, when the IDS needs to frequently retrain to handle new types of network attacks

Recommendation for Future Work

- Further analysis of each feature on a per outcome basis can be performed and used for additional processing in the feature-extraction module.
- The IDS considered is a signature-based IDS, which detects network attacks or intrusions through patterns in the features. Other approaches such as anomaly-based IDS can be considered to complement signature based IDS, where anomaly-based IDS detects behavioral deviations from normal network behavior.
- Optimize the IDS can be configuration for the number of sub-NNs present in a parallel NN implementation, threshold values to remove unneeded features, number of neurons in the hidden layer, number of hidden layers, testing-to-validation ratio used for NN training and threshold values to determine unknown outcome types.

Reference

[1] M. A. Hogo, “Temporal analysis of intrusion detection,” in Int. Carnahan Conf. on Security Technol., 2014, pp. 1–6.

NUS
National University of Singapore