CSCI-735 Project Phase 3: Result Analysis

Parijat Kawale
pk7145@rit.edu
Rochester Institute of Technology
Rochester, New York, USA

Archit Joshi
aj6082@rit.edu
Rochester Institute of Technology
Rochester, New York, USA

1 WHAT IDS DESIGN AND EVALUATION ASPECTS HAVE YOU IDENTIFIED IN YOUR RESEARCH?

In the first phase of the project, we worked on two different types of IDS. The first IDS design was for a misuse-based IDS. The initial approach was to use Suricata[7], Snort[6] or a similar existing IDS tool to identify the attack types. However, while working on the custom data, Suricata was not able to read through the data. So, after careful analysis and research, we decided that the design of a Support Vector Machine Classifier (SVM-Classifier)[5] was the most appropriate to design and detect misuse-based attacks. The design of the Support Vector Machine Classifier is as follows:

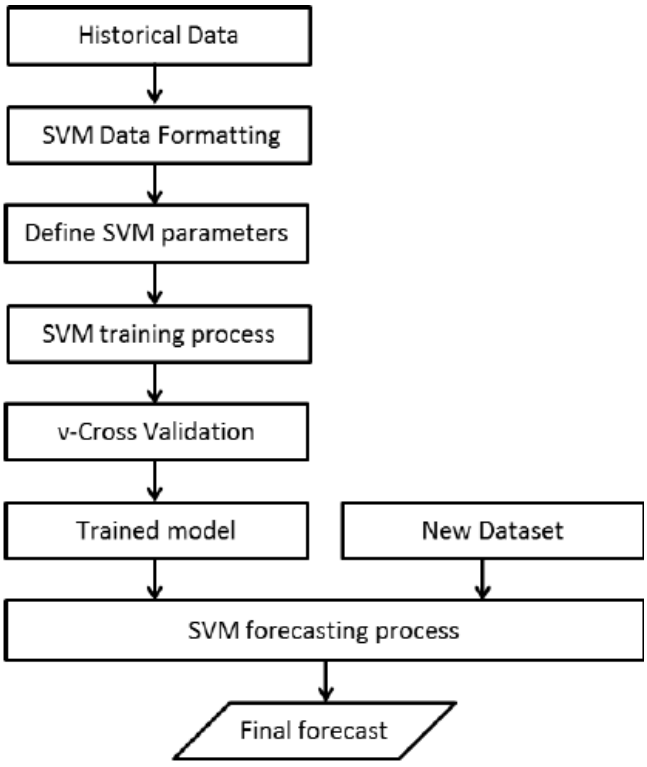

Figure 1: SVM design for misuse-based classifier[8]

As you can see from the design, the process of classifying using SVM starts with pre-processing of the data. After preprocessing, we defined the hyper-parameters for the algorithm and trained the model to begin forecasting on the test data-set.

Conference'17, July 2017, Washington, DC, USA
2023. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

The second IDS design was for an anomaly-based IDS. Similar to misuse-based IDS, we tried to work with existing IDS tools. However, since we are working on custom data, we decided to work with Isolation Forest Model[2] after careful analysis. The design for Anomaly-based IDS is as follows: As you can concur from the design,

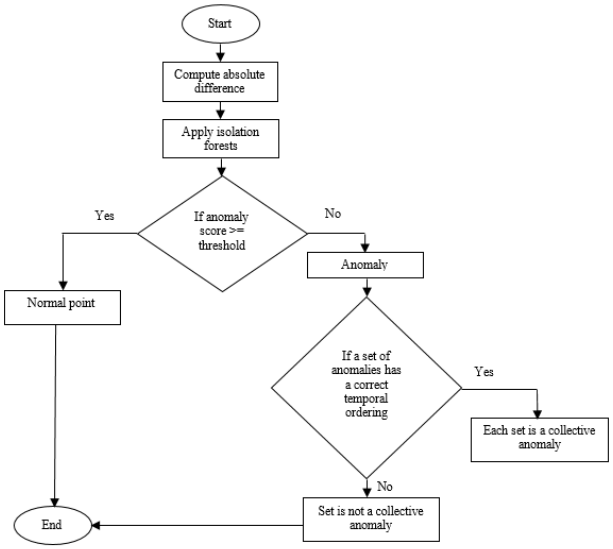

Figure 2: Isolation Forest design for anomaly-based classifier[1]

the process of identifying the attacks using Isolation Forest Model starts with data preprocessing (defining what a normal vs anomaly label is) followed by defining hyper-parameters and training the model. Finally, the trained model is tested on unknown data to check for its ability to classify attacks.

The evaluation aspect for the misuse-based classifier was the false positive rate and false negative rate for the test data-set. For the anomaly-based classifier, the evaluation aspect we identified was the accuracy of the model to identify if the packet is normal or an attack i.e an anomaly.

For phase 2 of this project we studied how we can implement Artificial Neural Networks (ANN) in the IDS design process. ANNs give significant advantage over classifiers as they can capture complex relationships and patterns in data, including non-linear relationships, better than many traditional classifiers. They are composed of interconnected nodes (neurons) in layers, allowing them to model intricate relationships within the data. They can also automatically learn relevant features from the input data during the training process. Traditional classifiers often rely on manual feature engineering, where human experts identify and select relevant features.

ANNs, especially deep learning models, can learn hierarchical representations of features, eliminating the need for explicit feature engineering in many cases.

The Misuse IDS uses a Sequential model from the Tensorflow Keras[10], which is a feedforward neural network and the data flows in one direction from input to output. The model has two hidden layers. The first layer of the hidden layer consists of 64 neurons. It is aimed at providing enough capacity to learn complex patterns without being too large to cause overfitting or excessive computation. The second layer has 32 neurons that funnel the network. We use the ADAM[9] optimizer for the learning rate in the model and sparse categorical cross-entropy loss function which is suited for multi-class classification. For the anomaly-based IDS, the architecture remains the same.

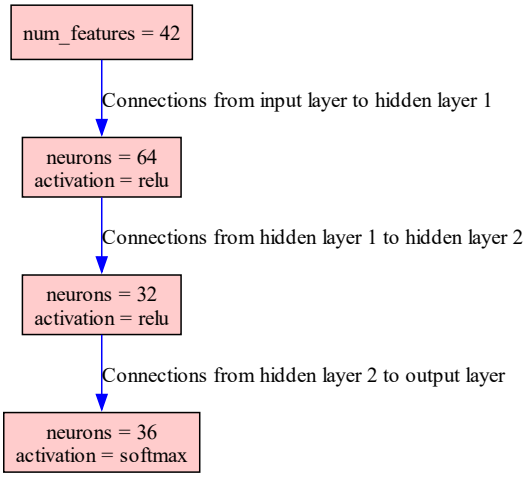

Figure 3: Neural Network architecture flowchart for 42 features and 36 labels

Throughout the design phase the following objectives were put on the forefront - a simple enough model that can be implemented and understood by the user easily and model that avoids over-fitting the data to give us good accuracy.

2 WHICH IDENTIFIED ASPECTS DO YOU CONSIDER THE MOST IMPORTANT?

For phase one, we used Support Vector Machines and Isolation Forest models to develop an IDS that would have good accuracy. These models were able to give us results including multiple metrics, that is, we had an accuracy of 96% and 80% for misuse-based and anomaly-based classifiers respectively.

In phase two, we used two different architectures of an Artificial Neural Network model to develop a misuse-based IDS and anomaly-based IDS. The accuracies observed by using neural networks were 96.65% and 96.34% for misuse-based and anomaly-based IDS respectively.

Through this process, we identified that the most important aspect of the misuse-based classifier is the false positive rate and false negative rate. For anomaly-based IDS, we identify that the most important aspect is the accuracy of the model after training on testing data.

3 HOW DO YOU ADDRESS THESE ASPECTS AND PROBLEMS IN YOUR PROJECT?

3.1 Aspects of the project

For the aspects of the project, we will discuss the misuse-based IDS followed by anomaly-based IDS.

For misuse-based IDS, the most important aspect we found was the ratio of the false positives to false negatives. The false positives are calculated using following formula:

$$False\ Positive\ Rate = \frac{FP}{Actual\ Negative} = \frac{FP}{TN+FP}$$

Figure 4: False Positive Ratio / False Positive Rate

The false negatives are calculated using the following formula:

$$False\ Negative\ Rate = \frac{FN}{Actual\ Positive} = \frac{FN}{TP+FN}$$

Figure 5: False Negative Ratio / False Negative Rate

For anomaly-based IDS, the most important aspect is the accuracy of the model to identify if the current packet or request is normal or an attack. The accuracy is calculated using the formula:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

Figure 6: Accuracy

3.2 Problems of the project

The initial issue we observed was that we were working with stale data which limits the usage of pre-existing IDS tools like Snort[6] and Suricata[7]. As these tools work with live data it was difficult for us to simulate the scenario and adjust the data we have to these tools. Thus we went ahead with building our custom machine learning and AI models to analyse the data and predict intrusive behaviour. The next major issue that we encountered was the quality of the data. The data was unclean and semi-structured with categorical and numerical attributes. The range of values for different numerical attributes of the data set was spread out. We normalized them to fit them within the range of 0 to 1. The categorical values had to be one-hot-encoded to tailor them to classifier models and multi-class classification problems when designing a misuse-based IDS.

4 COMPARE YOUR DESIGN AND EVALUATION RESULTS IN PROJECTS 1 AND 2. WHAT WOULD BE YOUR PREFERENCE IF YOU DO IT AGAIN?

For anomaly-based IDS, we would prefer using a Neural network-based IDS design for better results. This claim is supported by the evaluation results mentioned above. ANNs are better at handling large scale inputs and they are able to understand complex relationships between the data points more effectively as they feature engineer and learn while training. Without implementing unsupervised / reinforcement learning this would be hard to achieve for the regular classifier IDS. As seen in Figure 7 the accuracy rate for the normal classifier misuse IDS remains the same over multiple epochs as the model isnt capable of learning that much in the process. However the curve looks different for the ANN models in Figure 8 as ANNs are capable of self learning over multiple epochs.

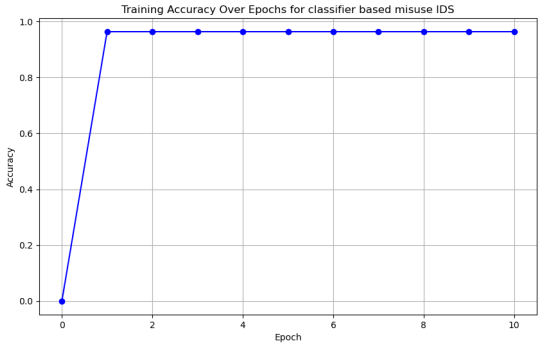

Figure 7: Accuracy over multiple epochs for classifier based misuse IDS

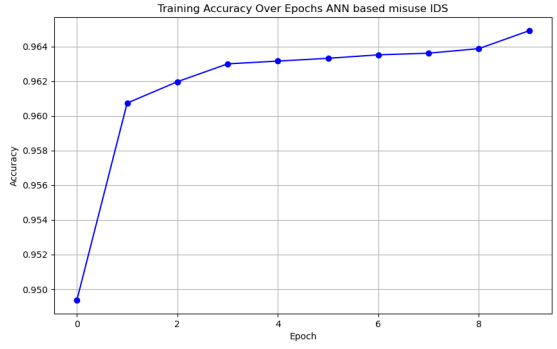

Figure 8: Accuracy over multiple epochs for ANN based misuse IDS

The false positive and false negative ratios along with the accuracies for different models are summarized in the table below. This would corroborate with our findings discussed previously. Although implementing ANNs would require a little bit more knowledge and expertise, they are more effective, avoid over-fitting and are able to learn more from the training data to realise complex relationships in the data.

IDS Type	Method	FPR	FNR	Accuracy
Misuse Based	SVM	NA	NA	96.365%
Misuse Based	ANN	0.0010	0.036	96%
Anomaly Based	Isolation Forests	NA	NA	80.518%
Anomaly Based	ANN	0.0009	0.0322	97%

Table 1: Comparison of IDS Performance Metrics - Accuracy, False Positive Ratio (FPR), and False Negative Ratio (FNR)

5 WHAT NOVEL AND INTERESTING IDEAS IN IDS DESIGN HAVE YOU IDENTIFIED, ADDRESSED, AND/OR SOLVED IN YOUR PROJECT?

Following are some of the interesting ideas in IDS design that we have identified/ addressed/solved in our project:

5.1 Handling Imbalance data

The initial observation while implementing the IDS designs was that the distribution of testing and training data was unbalanced. To tackle this issue, we used the parameters from the sklearn library - train_test_split() method, stratify parameter - that would ensure that the proportion of training and testing data is relatively relevant. The stratify parameter ensures the labels are distributed uniformly so that the models are trained efficiently for all data labels.

5.2 Integration of Two Detection Strategies

The implementation of the IDS has two different strategies, that is, misuse-based and anomaly-based detection. This dual approach allows the IDS to detect known attack patterns (misuse detection) and identify unusual activities that could indicate new, unknown attacks (anomaly detection) that cause out-of-the-ordinary anomalous behaviour.

5.3 Feature Engineering and Data preprocessing

The IDS design robustly handles data. We have scaled the data using the MinMaxScaler[3] so that the features are normalized within a range. Also, we have included usage of OneHotEncoder[4] which encodes the categorical variables. These measures ensure that the data is in the correct format for training and that the machine learning model has faster performance.

6 WHAT PROBLEMS HAVE YOU IDENTIFIED BUT NOT SOLVED? WHY?

While working through phases 1 and 2 for the IDS design, we experimented on misuse-based IDS for its ability to identify a previously unknown attack. We essentially removed two labeled attacks from the training data-set and then tested the misuse-based model on

the rest of the data. However, while testing, we added those two attacks which the model was not trained on. This gave unexpected results as the misuse-based IDS was not able to identify these new attacks, that is, for some instances of the data the output was correctly identified as an attack for some instances it was classified as normal data. This problem would have been resolved through the usage of reinforcement learning or unsupervised however, this was not feasible to implement as this requires a deeper understanding of the domain and much more resources than in the scope of the project requirements.

7 DO YOU THINK YOUR DESIGN APPROACH REPRESENTS THE BEST WAY TO SOLVE OR ADDRESS THESE PROBLEMS? WHY YES OR NO?

Our design approach is better than most standard ML algorithms in order to tackle the problems mentioned in the previous questions. The reason for this is that we use a blend of machine learning models with carefully tuned hyper-parameters to give the most optimized and accurate results when compared to just picking up a normal ML algorithm. We also focus on data cleaning and normalization which is a very important step for any machine learning, AI and Data Science tasks.

However, we do not think that this would be the best approach for IDS Design as the current IDS design is missing the capability of self-learning that is prevalent in the AI-based algorithm.

8 WHY DO YOU THINK YOUR TOOL IS THE BEST TO DESIGN AN IDS?

Few of the reasons why we think our tool is the best to design an IDS:

8.1 Multiple identification strategies

We use both anomaly-based IDS and misuse-based IDS designs to identify if the data qualifies as an attack or normal data and if it is an attack then identify the type of attack. This will help in better classification of the data enhancing the accuracy of the data and providing the end user with the details of the type of attack.

8.2 Robust Evaluation Metrics

Our tool provides the end user with a plethora of evaluation metrics starting from accuracy, confusion matrix to precision, recall, and f1 score. This ensures that users can use different metrics for further analysis if required. Also, it gives a clearer picture in terms of how the model has performed overall and an idea towards what parameters might require adjustments.

9 WHAT LIMITATIONS DOES YOUR DESIGN HAVE?

The major limitation that we have in our design is discussed below:

9.1 High Dependency on labeled data

Our model depends very highly on availablity of large quantities of well labelled data. The more data we have the better trained the models will be. But the models dont implement self learning and unsupervised learning capabilities. In realtime scenarios new attacks are developed every single day and it may not be feasible to expect large quantities of labelled data with these up-to-date attack examples. This was visible when we removed 2 labels during training and added them during testing; the model was not able to generalize and the accuracy dropped down by almost half.

9.2 Need for preprocessing data

The data that is taken into consideration needs to be processed in a particular manner so that it can be used as input to the IDS design. Firstly the data needs to be normalized, that is the numerical data needs to be scaled and fitted in the range of 0,1. Once we have normalized the data, we need to encode the categorical variables present in the data. This encoding is not only helpful to enhance the performance of the IDS by simplifying the data in a manner that can be easy to read. This essentially means that the IDS design requires considerable preprocessing of the data before the actual training takes place.

10 WHAT ARE THE MOST IMPORTANT YOUR RESULTS?

- During phase 1, we developed a misuse-based IDS using the SVM algorithm and an anomaly-based IDS using the Isolation Forest Model. The results obtained during this phase are as follows:

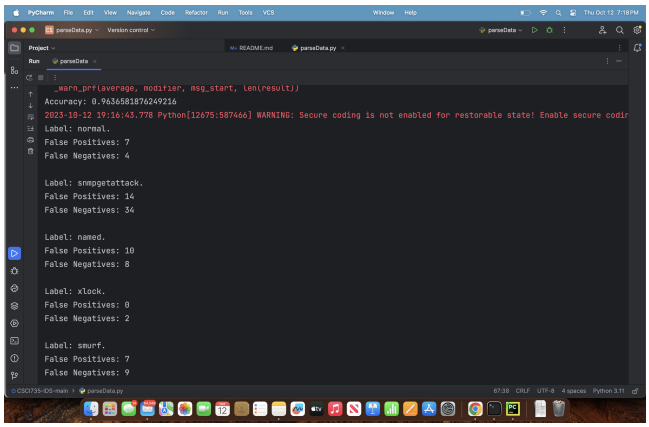

Figure 9: Console output from Misuse-based IDS using SVM model

During phase 2 of IDS design, we developed the misuse-based IDS and anomaly-based IDS using an artificial neural networks resulting in following results:

The most important result that we found from the analysis and development done during phases 1 and 2 is that the IDS based on artificial neural network performed better in all aspects when compared to other IDS designs.

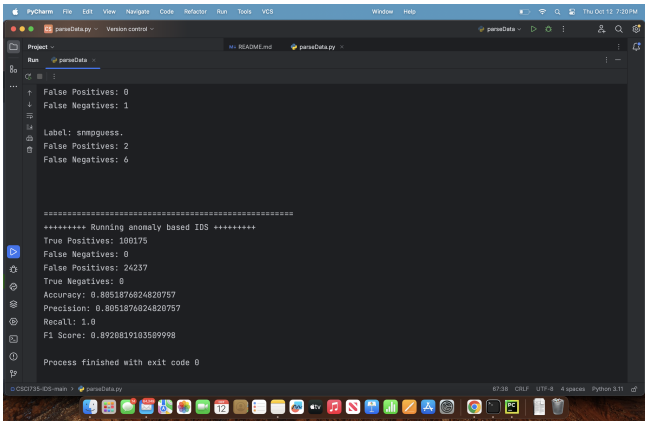

Figure 10: Console output from Anomaly-based IDS using Isolation Forest Model

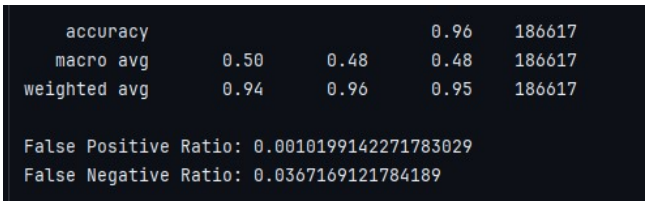

Figure 11: Console output from Misuse-based IDS with accuracy and FP+FN ratios

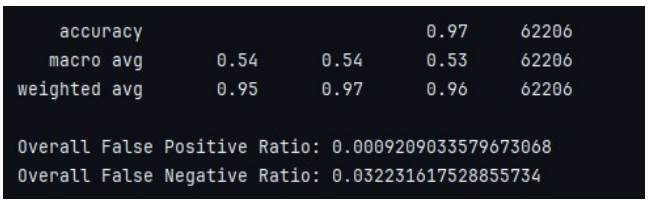

Figure 12: Console output from Anomaly-based IDS with accuracy and FP+FN ratios

11 WHICH RESULTS AND SOLUTIONS ARE NOVEL AND COULD BE PATENTED?

This project was our first foray into understanding intrusion detection systems and exploration towards how we can build one from scratch using machine learning and AI techniques. At this point although novel and easy to understand, our ideas aren't yet patent worth; as there is still much more room for improvement. In the future with significant experimentation and implementing unsupervised and reinforcement learning we can look into patenting future work.

REFERENCES

[1] [n. d.]. Iso Model Flowchart. https://www.researchgate.net/figure/Flowchart-of-proposed-algorithm-using-isolation-forests_fig2_339779712
[2] [n. d.]. Isolation Forests. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html
[3] [n. d.]. Min Max Scaler. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
[4] [n. d.]. One Hot Encoding. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html
[5] [n. d.]. Sklearn SVM. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
[6] [n. d.]. SNORT IDS tool. https://www.snort.org/
[7] [n. d.]. Suricata IDS tool. https://suricata.io/
[8] [n. d.]. SVM Model Flowchart. https://www.researchgate.net/figure/Operation-Flow-Chart-of-the-SVM-Model_fig1_261040572
[9] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG]
[10] Tensor-flow. [n. d.]. https://www.tensorflow.org/guide/keras/