

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/288991542>

A Deep Learning Approach for Network Intrusion Detection System

Conference Paper in Security and Safety · December 2015

DOI: 10.4108/eai.3-12-2015.2262516

CITATIONS

280

READS

8,355

4 authors, including:



Ahmad Yazdan Javaid

University of Toledo

66 PUBLICATIONS 704 CITATIONS

[SEE PROFILE](#)



Mansoor Alam

National University of Sciences and Technology

84 PUBLICATIONS 2,156 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Network Security [View project](#)



LVC Enterprise [View project](#)

A Deep Learning Approach for Network Intrusion Detection System

Presented By:

Dr. Ahmad Y. Javaid

Co-authors:

Quamar Niyaz

Dr. Weiqing Sun

Dr. Mansoor Alam

Outline

- Introduction
 - Self-taught Learning (STL)
 - NSL-KDD
- Implementation of NIDS
- Results
- Conclusion

Outline

- **Introduction**
 - Self-taught Learning (STL)
 - NSL-KDD
- Implementation of NIDS
- Results
- Conclusion

Introduction

- NIDS can be categorized as:
 - Signature based NIDS (SNIDS)
 - Attacks signatures are pre-installed
 - Anomaly detection based NIDS (ADNIDS)
 - Deviation from normal traffic pattern is attack
 - Most common among research community

Introduction

- Challenges arise for developing an efficient ADNIDS
 - Proper feature selection
 - Organization's reluctance to report any intrusion
 - To maintain privacy of various users
- Deep Learning can help to overcome the challenges of developing an efficient NIDS

Outline

- Introduction
 - Self-taught Learning (STL)
 - NSL-KDD
- Implementation of NIDS
- Results
- Conclusion

Self-taught Learning (STL)

- A deep learning approach consists of two stages for classification
 - Feature representation learnt from large unlabeled data, i.e., Unsupervised Feature Learning (UFL)
 - Learnt representation is applied on labeled data
- Sparse auto-encoder used for UFL

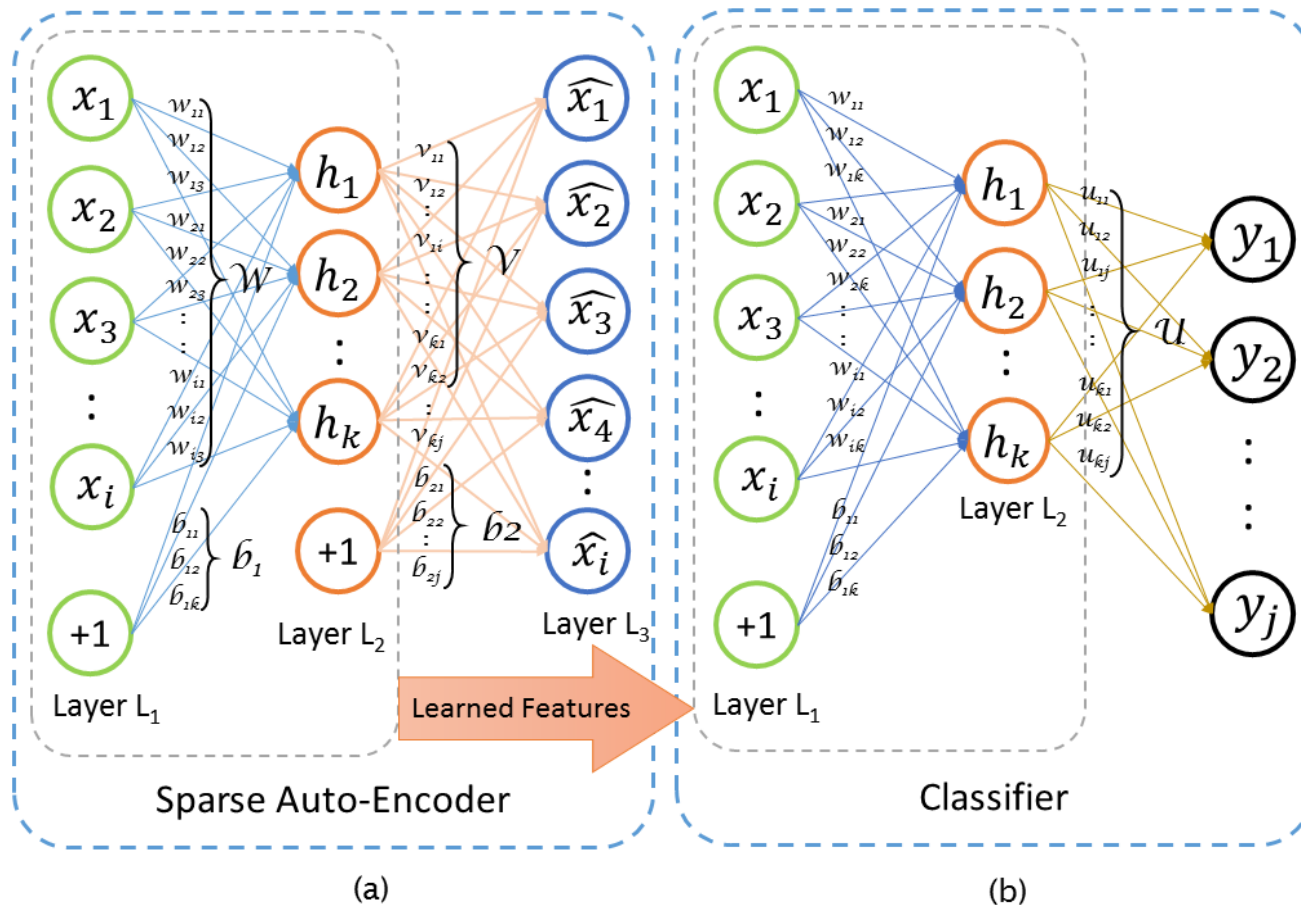


Figure 1: Two stages of Self-taught Learning (STL)

Outline

- Introduction
 - Self-taught Learning (STL)
 - **NSL-KDD**
- Implementation of NIDS
- Results
- Conclusion

NSL-KDD Dataset

- An improved version of KDD Cup 99 intrusion dataset
 - Eliminated redundant records in KDD Cup 99
- Dataset records consist of 41 features labeled with normal or a particular attack traffic
 - Includes basic features, traffic features accumulated in a window interval, and content features

NSL-KDD Dataset

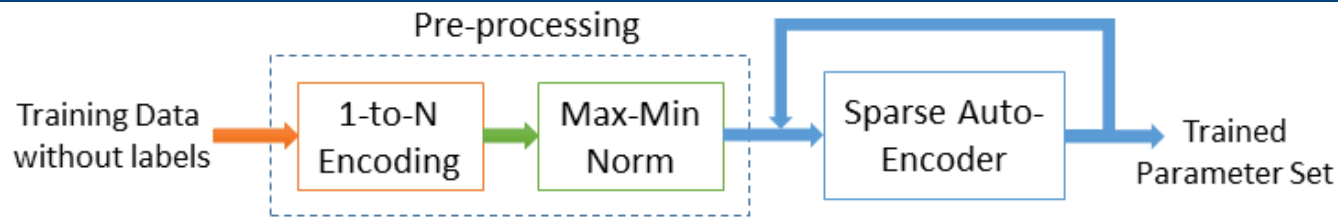
- Out of 41 features:
 - 3 nominal, 4 binary, and 34 continuous
- Training and test data contains 23 and 38 traffic classes including normal and attack traffic
 - Attacks grouped into 4 categories: DoS, Probing, U2R, and R2L

Outline

- Introduction
 - Self-taught Learning (STL)
 - NSL-KDD
- **Implementation of NIDS**
- Results
- Conclusion

NIDS Implementation

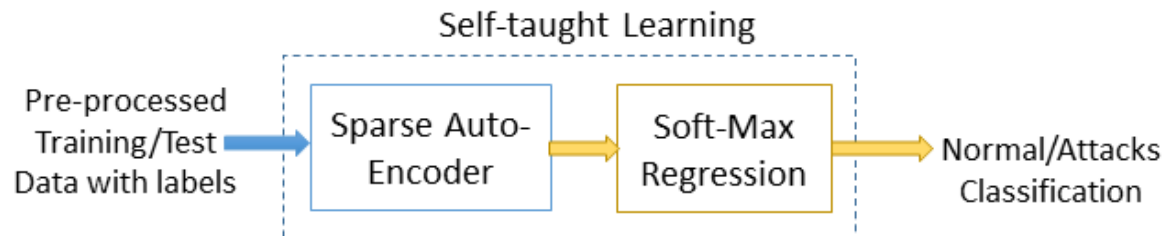
- Implemented using MATLAB/Octave
- Pre-processed the dataset before applying STL
 - 1-to-N encoding to convert nominal attributes to discrete attributes
 - Max-min normalization of the attributes
- Evaluated for both the training and test data



(a) Feature Learning from pre-processed data



(b) Soft-max Regression classifier training for the derived training data



(c) Classification using Self-taught Learning

Figure 2: Steps involved in NIDS Implementation

Outline

- Introduction
 - Self-taught Learning (STL)
 - NSL-KDD
- Implementation of NIDS
- **Results**
- Conclusion

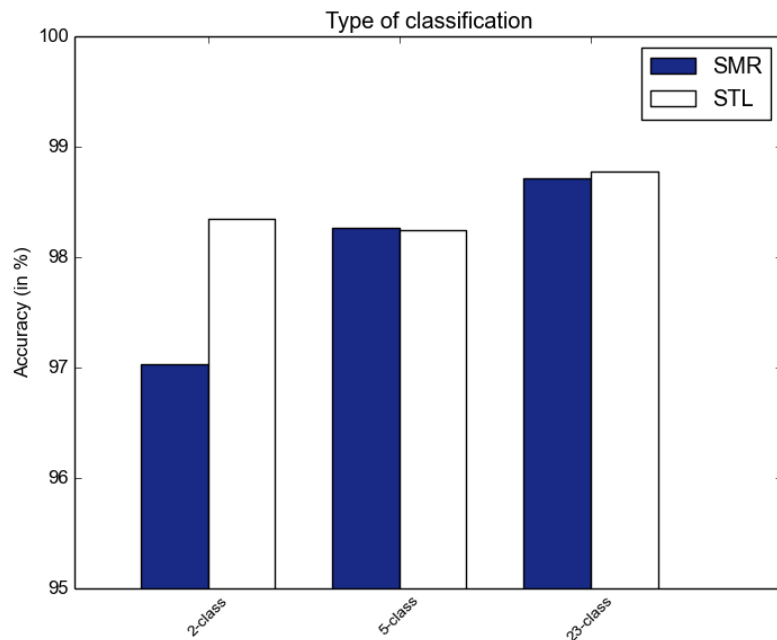
Accuracy Metrics

- **Accuracy** %age of correctly classified records
- **Precision** $P = TP / (TP + FP) * 100\%$
- **Recall** $R = TP / (TP + FN) * 100\%$
- **F-measure** $F = 2 * P * R / (P + R) * 100\%$

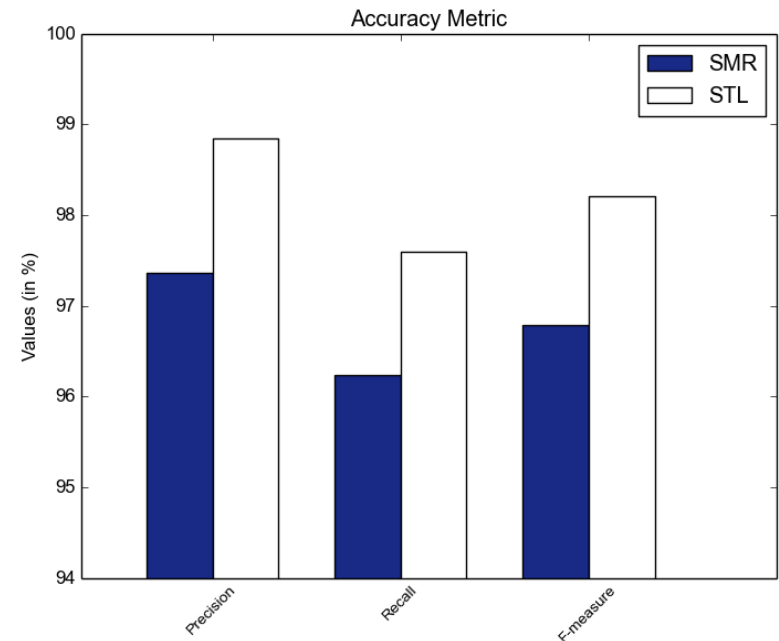
Performance Evaluation

- Implemented the NIDS for 3-types
 - Normal and Anomaly (2-class)
 - Normal and four attack categories (5-class)
 - Normal and 22 attacks (23-class)
 - For training data only
- Precision, Recall, and F-measures evaluated for 2-class and 5-class

Evaluation based on Training data

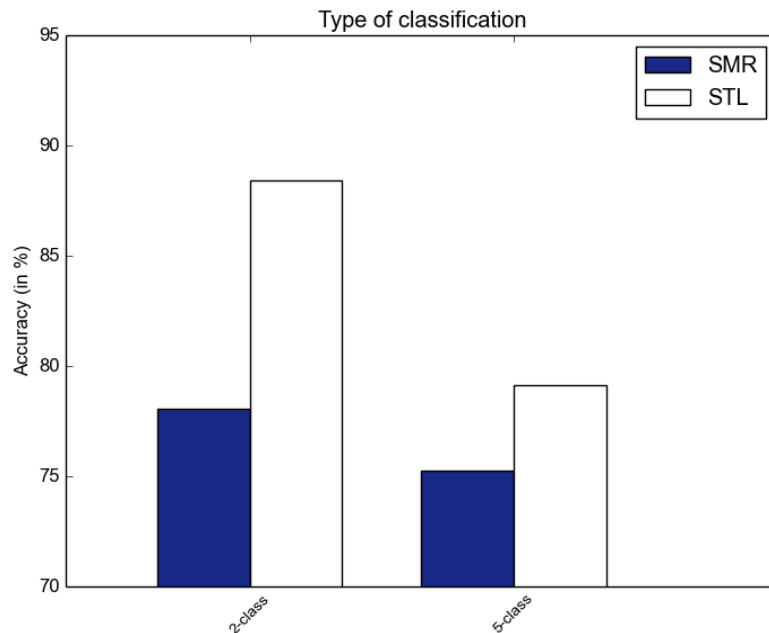


- Accuracy evaluated for 2, 5, and 23-classes
- STL achieved >98% accuracy for all types

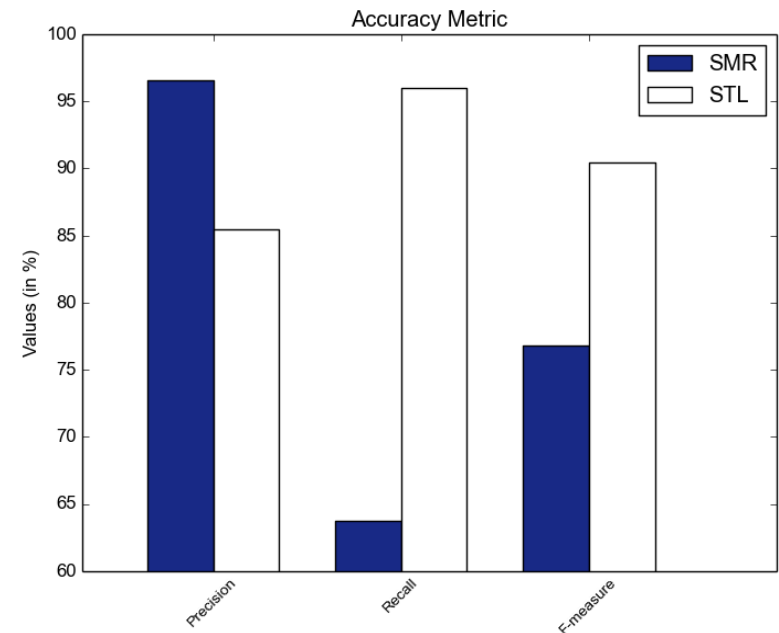


- Precision, recall, and f-measure evaluated for 2-class
- STL achieved f-measure value ~99%

Evaluation based on Test data



- STL achieved accuracy of ~88% for 2-class
- Better than various previous methods



- STL achieved ~90% f-measure value
- SMR achieved only ~77%

Outline

- Introduction
- Overview
 - Self-taught Learning (STL)
 - NSL-KDD
- Implementation of NIDS
- Results
- **Conclusion**

Conclusion

- STL based NIDS showcased good performance compared to other methods on NSL-KDD dataset
- Future work
 - Performance enhancement using other DL methods
 - To be implemented for real-time network operation

Thanks!

e-m@il: ahmad.javaid@utoledo.edu