Review of High-Performance Network Intrusion Detection Engine

Sampath C V Gowda[1], Affan Baig M[2], Vishaal V[3], Chandan R[4] and G M Harikrishnan[5]

Abstract:

As the proliferation of connected devices and services increases, so does the demand for protective measures against cyber-attacks. Intrusion Detection Systems (IDS) are a crucial component of network perimeter security, detecting attacks by inspecting network traffic packets or operating system logs. While machine learning techniques have shown effectiveness in intrusion detection, few have utilized the time-series information of network traffic data, and none have included categorical information in neural network-based approaches. In this paper, we propose network intrusion detection models based on sequential information using Long Short-Term Memory (LSTM) networks and categorical information using embedding techniques. Our experiments on the UNSW-NB15 dataset demonstrate significant performance improvements, with binary classification accuracy reaching 99.72%.

Keyword:.

Network Intrusion Detection , Machine Learning , Long Short-Term Memory, anomaly detection, UNSW NB15.

1 Introduction

Intrusion detection technology is an important guarantee for computer network security systems, which has been paid much attention by researchers in the field of network information security. Intrusion detection ensures the normal operation of computer network by collecting and analyzing various information and data on the network and identifying abnormal data. Intrusion detection system (IDS) is a "burglar alarm" in the field of computer security. Its purpose is to defend the system by combining the alarm issued when the network security is threatened with the inspection entity. Intrusion detection is a good complement to firewall which improves the system's ability to deal with network attacks, reduces economic losses for enterprises, and provides users with better services.

Networks play an important role in our current life, using the network we transfer data's easily while transferring data's we can face many security threats to avoid any vulnerabilities we use cyber security. Cyber security is a technique which prevents anonymous attacks in networks like anti-virus software, firewalls etc. But they are not strong enough to detect a new type of attack. To improve the network security Intrusion Detection System(IDS) is introduced. IDS used to detect, monitor and analyze any vulnerabilities for both software and hardware running during a network.

There are two approaches for detection of malicious network activities: Traditional methods and machine learning based ones . Both of them involve feature extraction stage, but differ in how to identify malicious activity. In the feature extraction approach, individual packets in network activities are summarized into high level events such as sessions . Each summarized record consists of feature values that characterize the high level event . Then, in traditional development of NIDS, security experts identify patterns of attacks, deciding threshold ranges for each features . On the other hand, in the machine learning based approach, a given model automatically learns patterns of malicious activities from a given dataset . Recently, machine learning based methods have been attracting more attention over traditional methods, due to its potential capability to detect more complicated patterns in a large scale dataset.

While many researchers have experimented various machine learning techniques, time-series information of network traffic data have not received much attention .As network activity occurs in timely manner, usage of sequential information in machine learning models should lead to more comprehensive analysis as long as the model has enough computational capacity for such additional information. Recurrent neural networks (RNNs) can capture temporal dependence in data, which brought significant advances in the fields of speech recognition and machine translation, and long short-term memory (LSTM) or gated recurrent unit (GRU) are popular RNNs . In addition to temporal dependence, categorical information has been neglected in neural network based NIDS. Categorical information means non numeric (or symbolic) features like protocol type, state, and service in network traffic data. While such features are crucial in recognizing malicious pattern activity, traditional neural network approaches could not accept them as input. Categorical features are very common in natural language processing (NLP), because words are symbols, and there are several feature embedding (or word embedding) techniques to handle symbolic words in NLP tasks, like language model and neural machine translation.

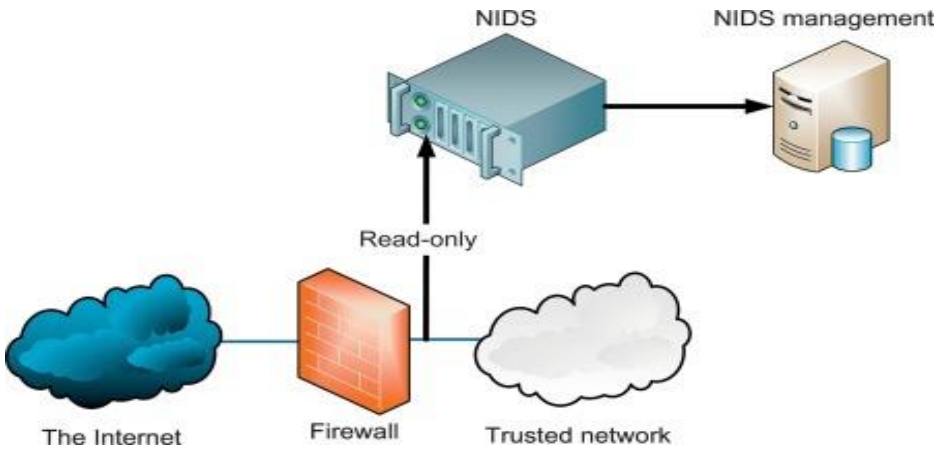

Figure 1 . Overview of Network Intrusion Detection System.

The Figure 1 shows the overview of intrusion detection: Here the firewall acts as Intrusion Detector; it stands between networks and filters traffic that might be unhealthy. Network Security can be monitored by administrators in network and security officers to provide a protected environment for user accounts, their online resources, personal details and passwords.

IDS can be divided into two types by their approach:

1. Signature Based Detection: It always uses signatures from previous data to detect intrusion; it may not be effective for new types of attacks.

2. Anomaly Detection: It uses an unusual pattern to detect attacks. Here we use Anomaly detection for IDS.

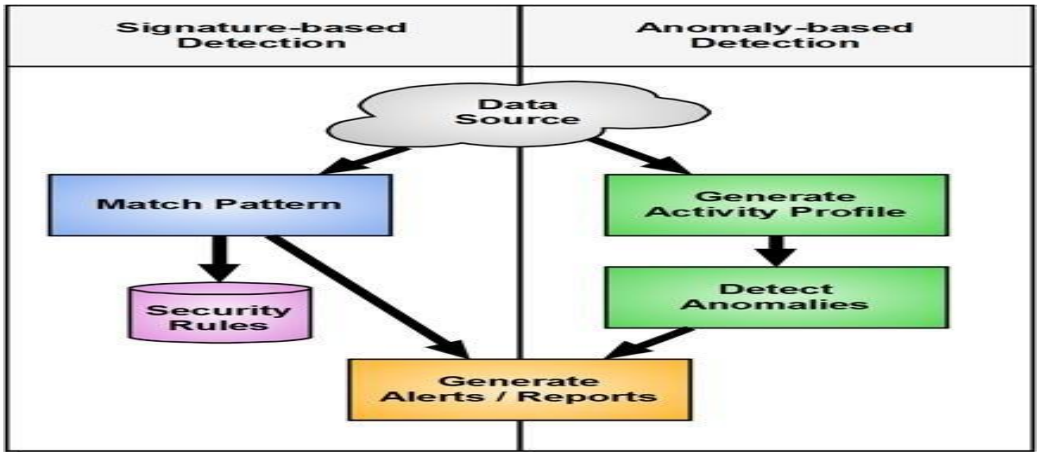

Figure 2 . SIGNATURE-AND ANOMALY-BASED IDS.

Attackers may act in two ways to try out their attacks in networks;

1) They create unavailability of network service for users.

2) Violating their personal information on the network.

To evaluate our proposed method for network intrusion detection system, we adopted the UNSW-NB15 dataset. UNSW-NB15 is an open dataset published by UNSW, a university in Australia, for network intrusion detection research in 2015. The KDD Cup 99 dataset used to be extensively used for network intrusion detection research in the past, but more recently UNSW-NB15 has been used because KDD Cup 99 does not contain much of the recent network hacking patterns.

There are many types of attacks in networks:

Denial of Service (DoS) is one among the frequent cases of attacks on network resources. It makes the network unavailable for users and creates traffic to crash their system. There are different sorts of DoS attacks, and each type has its own behavior by intruding network resources of users for their own purpose which is to render the network unavailable for its users.

Remote to Local (R2L) is one type of network attack, attackers send some type of files to gain unauthorized access to enter victim machines.

User to Root (U2R) attacks are similar to r2l attacks; it enters the root machine illegally to crash the local machines.

Probing is another type of attack in which intruders scan victim devices to find any weak spot in their machine to gain illegal access for their future attacks .There are many examples that represent probing over a network, like nmap, portsweep, ipsweep.

Attack Type	Attack Pattern
DoS	back, land, neptune, pod, smurf, teardrop, apache2, mailbomb, processtable, udpstorm
R2L	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster, snmpgetattack, named, xlook, xsnoop, snmpguess, worm, sendmail
U2R	Buffer_overflow, loadmodule, perl, rootkit, ps, sqlattack, httptunnel, xterm
Probe	Ipsweep, nmap, portsweep, satan, mscan, saint

Figure 3. Different Classes of Attacks

A NIDS analyzes the data packets that are transmitted over your business's network to identify possible cyber-attacks or malicious activities. While both systems monitor your private information networks looking for suspicious activity, they do it in different ways.

NIDS works by examining a variety of data points from different sources within the network. Packet headers, statistics, and protocol/application data flows are analyzed to determine whether malicious or anomalous activity has taken place. It can be used to identify possible security breaches on a system including sniffers and attacks on services such as HTTP/S, SMB, SSH etc.

Software to detect network intrusion and protect the network from unauthorized users. Intrusion learning models are used to create a predictive model to prevent attacks and it distinguishes the connection as “good” for normal and “bad” for intrusion.

As an open dataset, we use the UNSW-NB15 dataset, which is a broad-gauge network intrusion detection dataset. UNSW-NB15 was created for standardized evaluation of NIDS . Especially, it aimed to replace the KDD Cup 99 and NSL-KDD datasets, which have been popular datasets for NIDS over the years, but do not convey newly emerging network attack behaviors. As specified in, in order to reflect contemporary hacking behaviors, attacks in UNSW-NB15 were generated using IXIA Perfect Storm, which can simulate attacks listed in CVE website. After arranging a testbed environment with the attack generator, traffic was captured by TCP dump. Then the final dataset was formulated by conducting feature extraction with tools such as Bro and Argus.

Network intrusions have patterns according to their types. Generally, those patterns do not appear in a single packet, but can be dispersed for multiple packets. However, most of the previous machine learning methods for NIDS failed to address such characteristic, and they were not able to capture patterns that appear in multiple packets. For example, multi-layer Perceptron (MLP) performs intrusion detection with only one packet ignoring temporal dependency. Actually, if you want to detect DoS attack with MLP, it would be very tough because DoS is an attack to bring down a server by sending many packets, each of which is not very different from normal packets. This issue might not be limited to DoS attack, but also for other attack types. For more accurate intrusion detection, therefore, it is necessary to deal with multiple packets rather than a single packet.

2 Literature Review

Survey of literature work done by other researchers. I’ve learned some existing techniques from their research work, few of them are discussed below in the following table.

Authors	Methods	Remarks
Coelho et al [1]	They used homogeneity of data cluster and label to form a semi supervised data for feature selection . During a context of few labeled and many unlabeled instances	This method enhanced the performance of the feature selection process.
Gharaee and Hossein [2]	Proposed a genetic algorithm and SVM with a new feature selection technique to improve the IDS. They performed their work on KDD CUP 99 and UNSWNB 15 dataset.	The new feature selection method based on a genetic algorithm with innovative fitness function to increase the true positive rate and reduce the false positive simultaneously reduces the time taken for execution.
Gul and Adali [3]	Proposed a feature selection process for Intrusion Detection. Feature selection is an important process before classification is performed.	When selecting the important feature it will reduce the execution time and increase the accuracy of the model.
Zhang and Wang[4]	Proposed an effective wrapper based feature selection to increase the accuracy of the algorithm.	The wrapper method feature selection is based on Bayesian Network classifier.
Moustafa et al [5]	Compared the signature based network intrusion detection that Anomaly based detection is more efficient. The Authors evaluate their classification algorithm with two benchmark datasets of Network Intrusion Detection System (NIDS) NSL-KDD and KDD99 and find out that the datasets may be lacking in accuracy because of poor recent attack types, so the author used UNSW NB15 dataset.	The author shows that evaluation of UNSW NB15 is done in three aspects to find its complexity. Also this system offered a higher accuracy based on optimization in real time.

Selvakumar et al [6]	Proposed a novel intelligent intrusion detection for multi-class classification data. They have used the KDD CUP dataset. The dataset is preprocessed and FR algorithm is applied to get best features for classification.	They got 99.7% accuracy for intrusion detection. Compared with existing models they achieved a high accuracy rate.
Primartha and Tama [7]	Used three different (UNSW NB15, GPRS, and NSL-KDD) datasets to perform classification process using Random forest, Naive Bayes, and Neural Network	They got High accuracy and low warning rate and also K-cross validation was done

Authors	Methods	Remarks
Belouch et al. [8]	Proposed a two-stage classifier supported RepTree algorithm and protocol subset for network intrusion detection. To gauge the performance of systems their approach, they used UNSW-NB15 and NSL KDD dataset.	The feature technique is used to reduce the get best features here they get 20 best features out of 40. They have achieved detection accuracy of 88.95% and 89.85% on the UNSW-NB15 and NSL-KDD dataset.
Dhanabal and Shantharajah [9]	used an NSL-KDD dataset and applied a different classification algorithm	detected the effectiveness of the classification algorithm in anomaly detection
Tama et al [10]	Proposed hybrid feature selection and two-level classifier ensembles algorithm to improve the IDS. They have used NSL-KDD and UNSW NB15 dataset to perform their algorithm.	In hybrid feature selection there are three methods(genetic algorithm, particle swarm optimization, ant colony algorithm) used to reduce the size of features in the datasets.
Selvakumar et al [11]	Proposed the FRNN approach to improve accuracy by reducing false positives in Wireless Sensor Network (WSN). They have used a traced	They have achieved 99.87% accuracy compared with existing models.

	dataset and applied Allen's interval algebra for preprocessing and selected important features using the Fuzzy algorithm.	
Vanthana et al [12]	proposed an optimal packet concept to increase the effectiveness in the intrusion detection. They used traced file data's.	They introduce an indexing technique to reduce complexity and increase the accuracy in network intrusion detection.
Dahiya and Srivastava[13]	Proposed a framework during which a feature reduction algorithm is employed for reducing the smaller features than applied the supervised data processing techniques on UNSW-NB15 network dataset	They applied the supervised data processing techniques on UNSW-NB15 network dataset for fast, efficient and accurate detection of intrusion within the Netflow records using Spark.
Osama Faker [14]	combined big data and IDS to create an efficient IDS for a large number of data's. Here, CICIDS2017 and UNSW NB15 datasets are used to perform the classification. homogenetic metrics are used to select the best feature for classification and there are three algorithms used for classification techniques are Deep Feed-Forward Neural Network (DNN), Random Forest and Gradient Boosting Tree.	They got a high accuracy rate and 5-fold cross validation is done on Machine learning models.

3 Advantages and Disadvantages:

Advantages are:

Feature Extraction: LSTMs can automatically learn and extract relevant features from the data, reducing the need for manual feature engineering.

Anomaly Detection: LSTMs are effective for anomaly detection, which is crucial in NIDS to identify deviations from normal network behavior.

Real-time Monitoring: NIDS operates in real-time, continuously monitoring network traffic for abnormal patterns or behaviors. This allows for prompt responses to potential security incidents.

Anomaly Detection: Many NIDS use anomaly detection techniques to identify deviations from normal network behavior. This approach helps in detecting previously unknown or zero-day attacks.

Logs and Alerts: NIDS generates logs and alerts when suspicious activities are detected. Security administrators can use these logs to investigate and respond to potential threats promptly.

Memory Retention: LSTMs can capture long-term dependencies and remember information over extended periods. This is beneficial for identifying sophisticated and complex attacks that unfold gradually.

Robust to Irregular Patterns: LSTMs can handle irregular patterns in data, making them robust in scenarios where attacks exhibit non-uniform behavior.

Forensic Analysis: NIDS aids in forensic analysis by providing a historical record of network events. This information is valuable for understanding the timeline and impact of security incidents.

Disadvantages are:

Computational Complexity: LSTMs are computationally expensive compared to simpler models, which can be a challenge in real-time NIDS applications, especially in high-speed networks.

Training Time: Training deep learning models like LSTMs can be time-consuming, requiring substantial computational resources and potentially limiting their practicality.

Signature-based Limitations: Some NIDS rely on signature-based detection, which may be ineffective against new or polymorphic malware that does not match known signatures.

Encryption Challenges: Encrypted traffic poses a challenge for NIDS, as it may not be able to inspect the content of encrypted communications, potentially allowing malicious activities to go undetected.

Difficulty in Tuning: Hyperparameter tuning for LSTMs can be complex, and finding the right set of parameters may require considerable expertise

Complexity and Maintenance: Deploying and maintaining NIDS requires expertise. Configuring rules, keeping signature databases up to date, and adapting to evolving threats can be complex and time-consuming.

Overfitting: LSTMs can be prone to overfitting, especially when dealing with limited datasets. Overfitting can lead to poor generalization performance on unseen data.

Cost: Implementing and maintaining an effective NIDS can be costly, involving hardware, software, and ongoing operational expenses.

4 Challenges

Sequence Length and Padded Data: Network traffic data is inherently sequential, and determining an appropriate sequence length for LSTM input can be challenging. Padded data or truncated sequences may affect the model's ability to capture long-term dependencies.

Handling Variable-Length Sequences: Network sessions vary in length, and LSTM models struggle with variable-length sequences. Deciding on a fixed length for input sequences may lead to information loss.

Class Imbalance: The UNSW-NB15 dataset has class imbalance issues, with a significant number of normal instances compared to attack instances. Training an LSTM model on imbalanced data may lead to biased results.

Temporal Patterns in Attacks: LSTMs are sensitive to the temporal order of data. Certain attacks may exhibit subtle temporal patterns that are challenging for LSTMs to capture effectively.

Limited Interpretability: LSTMs are often considered as "black-box" models, making it challenging to interpret the decision-making process. Understanding the features and patterns learned by the LSTM is crucial for effective intrusion detection.

Hyperparameter Tuning: LSTM models involve tuning several hyperparameters, such as the number of hidden layers, number of LSTM units, and learning rates. Finding the optimal configuration can be time-consuming.

Resource Intensiveness: Training deep LSTM models on large datasets like UNSW-NB15 can be computationally expensive and may require significant computational resources.

Generalization to New Attacks: LSTMs trained on historical data may struggle to generalize to new and unseen attack patterns. The model's ability to adapt to emerging threats is crucial for real-world

deployment.

Noise in Network Traffic: Network traffic data often contains noise, and distinguishing between normal variations and actual attacks is challenging. LSTMs may capture noise as part of the learning process.

Training on Limited Data: Anomaly detection with LSTMs benefits from a large amount of labeled data, but obtaining a large number of labeled attack instances for training can be challenging.

Feature Engineering: LSTMs are designed to automatically learn features, but engineering informative features for network traffic data remains important for model performance.

5 Problem Statement

Develop a network intrusion detection system with high network throughput. The system should scan, classify and monitor the network traffic in real time without affecting the network throughput. Following are the features - Real time traffic analysis - Protocol analysis - Content searching - Detect variety of attacks and probes.

6 Problem Solutions

Creating a high-throughput Network Intrusion Detection System (NIDS) with real-time traffic analysis, protocol analysis, content searching, and detection of various attacks and probes involves a multi-faceted approach. Below is a conceptual outline for developing such a system. Keep in mind that the implementation details can vary based on your specific requirements, the technology stack you choose, and the characteristics of your network. Features of our NIDS system is shown in the table 6.

Table 6. Features of our NIDS

SI.No.	Feature	Description
01	Packet Capture	Use a high-performance packet capture library or tool to capture network traffic in real time. Tools like tcpdump or libraries like libpcap can be useful.
02	Real-Time Traffic Analysis	Implement a real-time traffic analysis module that can quickly process and analyze incoming packets. Use multithreading or asynchronous programming to handle simultaneous packet analysis efficiently.
03	Protocol Analysis	Develop protocol analyzers for common network protocols (TCP, UDP, ICMP, etc.). Implement protocol-specific parsers to extract relevant information and perform anomaly detection.
04	Content Searching	Develop content searching modules to inspect packet payloads for patterns, signatures, or malicious content. Utilize regular expressions, pattern matching, or specific content analysis techniques.
05	Attack and Probe Detection	Implement detection algorithms for various types of attack and probes. Utilize signature-based detection for known attacks and anomaly-based detection for detecting deviations from normal behavior.
03	Machine Learning for Anomaly Detection	Incorporate machine learning models for anomaly detection to identify patterns that may indicate malicious activity. Train models on labeled datasets, considering both normal and attack traffic.
04	Real-Time Alerts and Logging	Implement a real-time alerting system to notify administrators of detected intrusions. Log relevant information for forensic analysis and future improvements.
05	Optimizing for Throughput	Employ optimization techniques to minimize processing overhead. Implement parallel processing to distribute the load across multiple cores or nodes

7 Design and Methodology

Overall, our model incorporates three types of layers: embedding, LSTM, and fully connected layers. The embedding layer is utilized for nominal features, while continuous features are kept separate. Each nominal feature (proto, service, and state) is mapped to a respective vector of dimensions 5, 3, and 2 using the embedding layer. These output vectors are then concatenated with the continuous features and passed through the subsequent layers. The LSTM layer consists of a hidden state with 100 nodes, and the fully connected layer has 50 nodes with dropout. Leaky ReLU is applied as the activation function for non-linear transformation. For binary classification, an additional fully connected layer with 10 nodes is added. The dotted line indicates the layer working only in case of binary classification.

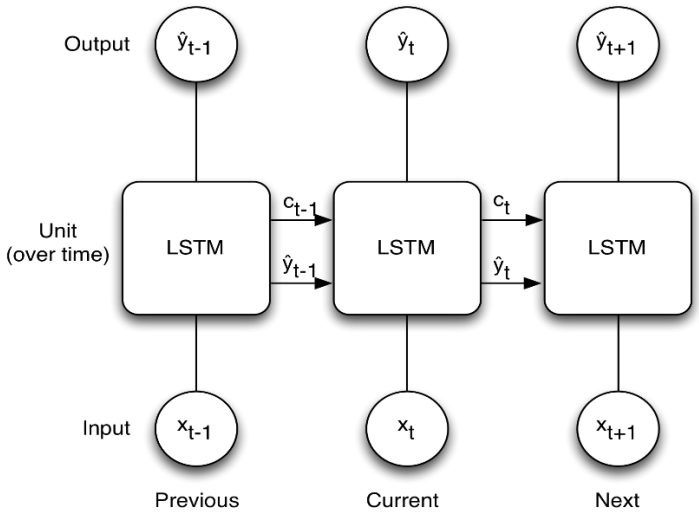

Fig. 2. Model Architecture: embedding, LSTM, and fully connected layers.

Methods for network intrusion detection: Evaluating rule-based methods and machine learning models on the unswnv15 dataset:

We have used Network intrusion detection is a task aimed to identify malicious network traffic. Malicious network traffic is generated when a perpetrator attacks a network or internet-connected device

with the intent to disrupt, steal or destroy a service or information. Two approaches for this particular task is the rule-based method and the use of machine learning. The purpose of this paper was to contribute with knowledge on how to evaluate and build better network intrusion detection systems (NIDS). That was fulfilled by comparing the detection ability of two machine learning models, a neural network and a random forest model, with a rule-based NIDS called Snort. The paper describes how the two models and Snort were constructed and how performance metrics were generated on a data set called UNSWNB15. It also describes how we capture our own malicious network traffic and the models ability to classify that data. The comparisons shows that the neural network outperforms Snort and the Random forest. We also present four factors that may influence which method that should be used for intrusion detection. In addition we conclude that we see potential in using UNSWNB15 to build NIDS based on machine learning.

Following UNSW-NB15's release, numerous studies have been conducted to
apply a wide range of machine learning methods to the given dataset. Suleiman et al. used a number of traditional machine learning algorithms, including Random Forest,

K-nearest neighbour and support vector machines, Gwon et al. The experiments showed that the J48 and K-NN algorithms were the most accurate and efficient models. Moustafa et al. experimented with a geometric area analysis approach utilising trapezoidal area estimation as an anomaly-based detection method.

A unique method for NIDS using a genetic algorithm and decision tree was put forth by Following the release of UNSW-NB15, numerous studies have been conducted to
use a wide range of machine learning methods on the given dataset. Using a variety of traditional machine learning techniques, including Random Forest, Suleiman et al.

K-nearest neighbour and Support Vector Machine, Gwon et al. The experiments revealed that the J48 and K-NN algorithms were the most accurate and efficient models. Using trapezoidal area estimation and geometric area analysis, Moustafa et al. experimented with an anomaly-based detection approach.

In the meantime, a novel method for NIDS using a genetic algorithm and decision tree was presented by Papamarztivanos et al. In their work, they generated detection rules that make up a decision-tree model using a genetic approach. After testing the final model on UNSW-NB15, it demonstrated good performance in identifying both attacks.

8 Result and Discussion

We test a wide range of training configuration combinations using feature embedding on LSTM. First, as previously mentioned, the LSTM model is trained in two different methods. One is learning from each output's errors (M2M), while the other is only learning from the final output's errors (M2O). Furthermore, we incorporate "multi-classification to binary-classification" (M2B) into binary classification. which
trains a multiclassification model that transforms all malicious labels and model outputs into a single labeled "attack." Eventually, each model is subjected to feature embedding (EMB).

Table 7. Case Studies with Real-world Examples

Authors & Year	Methods	Solution
Peter Manev (2020)[15]	Optimize Sensor Placement	The enterprise deployed Suricata, an open-source NIDS engine, across multiple locations to monitor and analyze network traffic.
Nelson advisors (2021) [16]	DarkTrace's AI technology	Darktrace, an AI-based NIDS, was implemented to analyze patterns and anomalies in real-time network traffic.
Vern Paxson (2018) [17]	Sits on a sensor and observes the network traffic	Zeek, an open-source network analysis framework, was deployed to monitor traffic in research and education networks.

9 Conclusion

In this paper, we proposed and experimented several IDS models based on LSTM and feature embedding. Evaluation was based on the UNSW-NB15 dataset which is suitable to reflect latest network traffic patterns. LSTM outperformed MLP with a significant margin (around 16% point or 13%) in accuracy and F1 score. Among LSTM models, the one with feature embedding was the best, since the embedding technique could capture categorical information which is crucial for attack recognition. We expect that real-time detection is possible in practice. Our future work includes making the model compatible with embedded system and Internet of things (IoT) by reducing the model complexity and shortening the necessary sequence length.

10 References

1. Greff, K., Srivastava, R.K., Koutn´ık, J., Steunebrink, B.R., Schmidhuber, J.: LSTM: A Search Space Odyssey. IEEE Transactions on Neural Networks and Learning Systems 28(10), 2222–2232 (2017)
2. Guo, C., Berkhahn, F.: Entity embeddings of categorical variables. CoRR abs/1604.06737 (2016), http://arxiv.org/abs/1604.06737 .
3. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human level performance on imagenet classification. CoRR abs/1502.01852 (2015), http://arxiv.org/abs/1502.01852 .
4. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation 9(8), 1735–1780 (Nov 1997). https://doi.org/10.1162/neco.1997.9.8.1735 .
5. Kim, J., Kim, J., Thu, H.L.T., Kim, H.: Long short term memory recurrent neural network classifier for intrusion detection. In: 2016 International Conference on Platform Technology and Service (PlatCon). pp. 1–5 (Feb 2016). https://doi.org/10.1109/PlatCon.2016.7456805 .
6. Laskov, P., Rieck, K., Muller, K.R.: Machine Learning for Intrusion Detection, pp. 366–373. IOS press (09 2008).
7. Mikolov, T., Kombrink, S., Deoras, A., Burget, L., Cernock´y, J.: RNNLM - Re- ˇ current Neural Network Language Modeling Toolkit. In: ASRU. pp. 1–4 (2011).
8. Moon, T., Choi, H., Lee, H., Song, I.: Rnndrop: a novel dropout for rnns in asr. In: ASRU. pp. 65–70 (12 2015). https://doi.org/10.1109/ASRU.2015.7404775 .
9. Moustafa, N., Slay, J.: Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS). pp. 1–6 (Nov 2015). https://doi.org/10.1109/MilCIS.2015.7348942 .
10. Moustafa, N., Slay, J., Creech, G.: Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks. IEEE Transactions on Big Data pp. 1–1 (2017).

https://doi.org/10.1109/TBDATA.2017.2715166 .

11. Nawir, M., Amir, A., Lynn, O.B., Yaakob, N., Ahmad, R.B.: Performances of machine learning algorithms for binary classification of network anomaly detection system. Journal of Physics: Conference Series 1018, 012015 (may 2018). https://doi.org/10.1088/1742-6596/1018/1/012015 .

12. Niyaz, Q., Sun, W., Javaid, A., Alam, M.: A deep learning approach for network intrusion detection system. EAI Endorsed Transactions on Security and Safety 3 (12 2015). https://doi.org/10.4108/eai.3-12-2015.2262516 .

13. Northcutt, S., Zeltser, L., Winters, S., Kent, K., Ritchey, R.W.: Inside Network Perimeter Security (2Nd Edition) (Inside). Sams, Indianapolis, IN, USA (2005).

14. Olah, C.: Understanding lstm networks (Aug 2015), https://colah.github.io/posts/2015-08-Understanding-LSTMs/

15. Papamartzivanos, D., Marmol, F.G., Kambourakis, G.: Dendron : Genetic trees driven rule induction for network intrusion detection systems. Future Generation Computer Systems 79, 558 – 574 (2018) https://doi.org/10.1016/j.future.2017.09.056 .

16. Pennington, J., Socher, R., Manning, C.D.: GloVe: Global Vectors for Word Representation. In: Empirical Methods in Natural Language Processing. pp. 1532–1543 (2014). https://doi.org/10.3115/v1/D14-1162 .

17. Shah, S.A.R., Issac, B.: Performance comparison of intrusion detection systems and application of machine learning to snort system. Future Generation Computer Systems 80, 157 – 170 (2018). https://doi.org/10.1016/j.future.2017.10.016

18. Staudmeyer, R.C.: Applying long short-term memory recurrent neural networks to intrusion detection. South African Computer Journal 56 (July 2015). https://doi.org/10.18489/sacj.v56i1.248 .

19. Suleiman, M., Issac, B.: Performance comparison of intrusion detection machine learning classifiers on benchmark and new datasets. In: 28th International Conference on Computer Theory and Applications (10 2018), https://iccta.aast.edu/

20. Tang, T.A., Mhamdi, L., McLernon, D., Zaidi, S.A.R., Ghogho, M.: Deep recurrent neural network for intrusion detection in sdn-based networks. In: 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft). pp. 202– 206 (June 2018). https://doi.org/10.1109/NETSOFT.2018.8460090 .

Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., AlNemrat, A., Venkatraman, S.: Deep learning approach for intelligent intrusion detection system. IEEE Access 7, 41525–41550 (2019). https://doi.org/10.1109/ACCESS.2019.2895334