symmetry

MDPI

Article

A Scalable and Hybrid Intrusion Detection System Based on the Convolutional-LSTM Network

Muhammad Ashfaq Khan [1], Md. Rezaul Karim [2,3] and Yangwoo Kim [1,*]

[1] Department of Information and Communication Engineering, Dongguk University, 30-Pildong-ro 1-gil, Jung-gu, Seoul 100-715, Korea; ashfaq_jiskani@dongguk.edu
[2] Fraunhofer Institute for Applied Information Technology FIT, 53754 Sankt Augustin, Germany; rezaul.karim@fit.fraunhofer.de
[3] Chair of Computer Science 5, RWTH Aachen University, 52074 Aachen, Germany
* Correspondence: ywkim@dongguk.edu; Tel.: +82-2-2260-3821

Received: 15 April 2019; Accepted: 17 April 2019; Published: 22 April 2019

Abstract: With the rapid advancements of ubiquitous information and communication technologies, a large number of trustworthy online systems and services have been deployed. However, cybersecurity threats are still mounting. An intrusion detection (ID) system can play a significant role in detecting such security threats. Thus, developing an intelligent and accurate ID system is a non-trivial research problem. Existing ID systems that are typically used in traditional network intrusion detection system often fail and cannot detect many known and new security threats, largely because those approaches are based on classical machine learning methods that provide less focus on accurate feature selection and classification. Consequently, many known signatures from the attack traffic remain unidentifiable and become latent. Furthermore, since a massive network infrastructure can produce large-scale data, these approaches often fail to handle them flexibly, hence are not scalable. To address these issues and improve the accuracy and scalability, we propose a scalable and hybrid IDS, which is based on Spark ML and the convolutional-LSTM (Conv-LSTM) network. This IDS is a two-stage ID system: the first stage employs the anomaly detection module, which is based on Spark ML. The second stage acts as a misuse detection module, which is based on the Conv-LSTM network, such that both global and local latent threat signatures can be addressed. Evaluations of several baseline models in the ISCX-UNB dataset show that our hybrid IDS can identify network misuses accurately in 97.29% of cases and outperforms state-of-the-art approaches during 10-fold cross-validation tests.

Keywords: intrusion detection system; deep learning; Spark ML; CNN; LSTM; Conv-LSTM

1. Introduction

Information and communication technologies now impact every aspect of society and people's lives, so attacks on ICT systems are increasing. Therefore, ICT systems need tangible, incorporated security solutions. The essential components of ICT security are confidentiality, integrity, and availability (CIA). Any activity trying to compromise CIA or avoid the security components of ICT is known as a network intrusion [1]. An intrusion detection system (IDS) is used for detecting such attacks. John et al. [2] published one of the first efforts on intrusion detection (ID) with a focus on computer security threat monitoring and surveillance. IDS is a kind of security management system utilized to observe network intrusions and nowadays is increasingly used in security systems [1,3]. An IDS typically monitors all inbound and outbound packets of a specific network to find out whether a packet shows signs of intrusion. A robust IDS can recognize the properties of maximum intrusion actions and automatically reply to them by sending warnings.

There are three main categories of IDS according to dynamic detection methods; the first is the misuse detection technique, which is known as a signature-based system (SBS). The second is the anomaly detection technique, which is known as an anomaly-based system. The third is based on a stateful protocol analysis detection approach [1,4]. The SBS depends on the pattern-matching method, comprising a signature database of identified attacks, and attempts to match these signatures with the examined data. When a match is found the alarm is raised, which is why an SBS is also known as a knowledge-based system. The misuse attack detection technique achieves maximum accuracy and minimum false alarm rate, but it cannot detect unknown attacks, while the behavior-based system is known as ABS and detects an attack by comparing abnormal behavior to normal behavior. Stateful protocol detection approaches compare the detected actions and recognize the unconventionality of the state of the protocol, and take advantage of both signature and anomaly-based attack detection approaches. In general, IDS is categorized into three types according to its architecture: Host intrusion detection system (HIDS), Network intrusion detection system (NIDS), and a hybrid approach [5,6].

A type of IDS in which a host computer plays a dynamic role in which application software is installed and useful for the monitoring and evaluation of system behavior is called a host-based intrusion detection system. In a HIDS event log files play a key role in intrusion detection [5,7]. Unlike HIDS, which evaluates every host individually, NIDS evaluates the flow of packets over the network. This kind of IDS has advantages over HIDS, which can evaluate the entire of the network with the single system, so NIDS is better in terms of computation time and the installation cost of application software on all hosts, but the foremost weakness of NIDS is its vulnerability to distribution.

A hybrid IDS, however, combines NIDS and HIDS with high flexibility and improved security mechanisms. A hybrid IDS joins the spatial sensors to address attacks that happen at a specific point or over the complete network. IDS can be classified into two main categories according to its deployment architecture: distributed and non-distributed architecture. The first kind of deployment architecture contains various ID subsystems over an extensive network, all of which communicate with each other, and is known as distributed deployment architecture; non-distributed IDS can be installed only at a single position, for example, the open-source system Snort [8]. As mentioned above, anomaly and misuse intrusion detection techniques have their limitations, but in our hybrid approach we combine the two techniques to overcome their disadvantages and propose a novel classical technique joining the benefits of the two techniques to achieve improved performance over traditional methods.

There are numerous conventional techniques for ID, for example access control mechanisms, firewalls, and encryption. These attack detection techniques have a few limitations, particularly when the systems are facing a large number of attacks like denial of service (DOS) attacks, and the systems can get a higher value of false positive and negative detection rates. In several recent studies, researchers have used machine learning (ML) techniques for intrusion detection with the ambition of improving the attack detection rates as compared to conventional attack detection techniques.

In our research, we first studied state-of-the-art approaches for IDS that apply ML techniques for ID. Then we proposed a novel approach to enhance performance in the ID domain [9]. However, simple ML techniques suffer from several limitations, while security attacks are on the increase. Upgraded learning techniques are required, particularly in features extraction and the analysis of intrusions. Hinton et al. [10] briefly explain that deep learning has achieved great success in various fields like NLP, image processing, weather prediction, etc. The techniques involved in DL have a nonlinear structure that shows a better learning capability for the analysis of composite data. The rapid progress in the parallel computing field in the last few years has also delivered a substantial hardware foundation for DL techniques.

Research has shown that a hybrid approach consisting of CNN and LSTM (aka, the Conv-LSTM network) shows a very powerful response and leads to high confidence in solving research problems such as video classification [11], sentiment [12], emotion recognition [13]; and in anomalous incident detection from a video [14]. Thus, to enhance the learning capability and detection performance of IDS, we propose a deep learning-based IDS system. In particular, we propose an improved version

of IDS, which is based on Spark ML and the Conv-LSTM network. While Spark ML-based classic machine learning models help identify anomalous network traffics, the Conv-LSTM network helps identify network misuses such that both global and local latent threat signatures can be addressed. As mentioned above, ABS and SBS both have a few limitations, but if we combine the two systems we can mitigate their drawbacks. We proposed a novel IDS joining the benefits of the two systems to improve performance as compared to traditional systems. The key contributions of this research can be summarized as follows:

- We proposed an attack detection method employing IDS, which is based on Spark ML and the Conv-LSTM network. It is a novel hybrid approach, which combines both deep and shallow learning approaches to exploit their strengths and overcome analytical overheads.
- We evaluated our IDS on the ISCX-UNB dataset and analyzed the packet capture file (pcap) with Spark; earlier researchers did not consider or evaluate raw packet datasets.
- We compare our hybrid IDS with state-of-the-art IDS systems based on conventional ML. The simulation results demonstrate that our IDS can identify network misuses accurately in 97.29% of cases and outperforms state-of-the-art approaches during 10-fold cross-validation tests.
- Our proposed IDS not only outperforms existing approaches but can also achieve mass scalability while meaningfully reducing the training time, overall giving a higher degree of accuracy with a low probability of false alarms.

The rest of this article is structured as follows: background on IDS and related works are discussed in Section 2. The proposed IDS framework with architectural and implementation details is covered in Section 3. Experimental results are demonstrated in Section 4, with a comparative analysis with existing approaches. Section 5 summarizes the research and provides some possible outlooks before concluding the paper.

2. Related Work

In the last three decades, numerous anomaly detection approaches have been proposed to develop effective NIDS, aiming at good predictive accuracy to perceive attacks and upgrading the network packet traffic's speed. These approaches vary from a simple statistical learning system to classic machine learning methods and recent deep learning-based approaches. Most of these approaches attempted to extract a pattern from the network so that attack traffic can be discriminated from regular traffic.

Existing ID systems are largely based on supervised learning methods, e.g., Support vector machines (SVM) [15–17], K-nearest neighbor (KNN) [18], Random forest (RF) [19,20], etc. However, these approaches produce many false alarms and have a low detection rate for attacks in IDS. Kim et al. [21] proposed a hybrid IDS framework that integrates anomaly attack detection with misuse attack detection using the C4.decision tree (DT) classification algorithm and SVM algorithm, respectively. They evaluated their hybrid IDS on the NSL-KDD dataset. Panda et al. [22] applied the Naive Bayes (NB) algorithm for anomaly detection, which is tested on the KDD Cup dataset and found to outperform many existing IDS in terms of the low false alarm rate and low computation time with low cost. Zaman et al. [23] used an enhanced algorithm called Support Vector Decision Function (ESVDF). Their IDS was evaluated on the DARPA dataset and found to outperform other conventional techniques.

Researchers also proposed other parallel and hybrid classification approaches by amalgamating the Self-Organization Map (SOM) and the C4.classifier [24]. In this approach, the SOM-based part was envisioned to regular model behavior, and any fluctuation from that usual behavior is identified as an intrusion. The C4.classifier-based part is used for misuse detection. This approach can be used to categorize those intrusion type data into the corresponding attack category, and the final decision was made by the module known as a decision support system (DSS). The DSS was evaluated from

every module by adding output and achieved maximum attack detection accuracy 99.8% on the KDD dataset along with a false alarm rate of 12.5% on the same dataset.

Albeit, the above approaches have shown good accuracy at detecting security threats to a certain degree, but it is essential to make some improvements, such as refining the accuracy and decreasing the number of false alarms [25–30]. Consequently, deep learning-based techniques are emerging, with the neural network (NN) [31] being at the core of these approaches as it provides very powerful responses not only for cybersecurity but also for other domains such as natural language processing (NLP), computer vision, and speech recognition [2,32]. Table 1 summarizes some related works.

Broadly, two fundamental characteristics account for DL-based approaches achieving tremendous success and effectiveness in these research areas: (i) hierarchical feature representations and learning capability; (ii) the ability to handle very high-dimensional data to extract valuable patterns. Previous approaches use shallow as well as deep learning techniques [32]. Gao et al. [33] proposed a restricted Boltzmann machine (RBM)-based deep belief network (DBN) to effectively learn data and identify unusual traffic from well-known datasets such as the KDD 99 dataset. Moradi et al. [34] generalized the ability of a multilayer perceptron (MLP) network analogous to layers in an attack, which is evaluated on the KDD 99 dataset. In the literature [2,35–37], LSTM-based deep learning approaches are proposed for feature selection and classification, which are evaluated on the KDD dataset.

These approaches are found to be very effective compared to the ML counterparts; researchers also proposed several ideas by combining ML- and DL-based approaches, aiming to develop robust IDS. For example, Mukkamala et al. [38] used a combined approach for classifying the connection records of the KDD 99 dataset, which is based on a support vector machine (SVM) and artificial neural network (ANN). While ANN learns the patterns of the data, the SVM is used for the classification. Javaid et al. [39] proposed a NIDS based on self-taught learning (STL). They applied their technique on the NSL-KDD dataset, and it outperformed previous approaches. Faraoun et al. [40] used multi-layered neural network backpropagation with K-means clustering for intrusion detection and experimented on the KDD 99 dataset.

On the other hand, the evolution of intrusion detection systems for the Internet of Things (IoT) is also an emerging research problem because the network traffic in real-time IoT-enabled devices is more pervasive and they are vulnerable to newer cybersecurity attacks [41]. Thus, researchers have focused on practical aspects such as mitigating the interference imposed by intruders in passive RFID networks [42].

Table 1. Overview of the state-of-the-art approaches.

Reference	Approach	Accuracy	Dataset
Yin et al. [15]	RNN IDS	90%	NSL-KDD
Reddy et al. [17]	SVM	99.95%	KDD99
B. Inger et al. [31]	ANN	99.67%	KDD99
Tsiropoulou et al. [42]	IMRA game theory	90.0%	Passive RFID
N. Gao et al. [33]	DBN	93.49%	NSL-KDD
Ghanem et al. [43]	Metaheuristic	96.4%	NSL-KDD
Sabhnani et al. [44]	MLP	97.0%	KDD99
Ying Chung et al. [45]	SSO	93.0%	KDD99
Kakavand et al. [25]	Ada boost + DT	97.0%	ISCX 2012
Kumar et al. [26]	PCA	94.05%	ISCX 2012
Yassin et al. [27]	AMGA2-NB	98.8%	ISCX 2012
Tan et al. [29]	MCA + EMD	90.12%	ISCX 2012
Sallay et al. [30]	PLL + NGL	95.30%	ISCX 2012

Note: The KDD 99 dataset contains 41 features of normal or attack types (denial of service (DOS), the user to root (U2R), remote to local (R2L), and probing attack). The NSL-KDD dataset is an improved version of the KDD 99 dataset. The ISCX 2012 dataset contains network traffic for seven days under practical and systematic circumstances.

Existing IDS systems that are typically used in traditional network intrusion detection system often fail and cannot detect many known and new security threats, largely because those approaches are based on classical ML methods that provide less focus on accurate feature selection and classification. Consequently, many known signatures from the attack traffic remain unidentifiable and become latent. With the fast development in the field of big data and computing power, DL techniques have blossomed and been used extensively in several fields, which is why network traffic is being generated at an unprecedented scale. This imposes a great challenge to existing IDS systems because these approaches are not only unscalable but also often inefficient. To address these issues and improve the accuracy and scalability, we propose a scalable and hybrid IDS, which is based on Spark ML and the convolutional-LSTM (Conv-LSTM) network.

3. Materials and Methods

In this section, we discuss the overall architecture of the proposed approach. First, we give an overview of the architecture, which will be followed by dataset preparation. Finally, we discuss the implementation details.

3.1. Architecture of the Proposed Hybrid IDS

As shown in Figure 1, our proposed IDS system comprises of two learning stages: (i) Stage 1 is employed for anomaly detection, which is based on classic ML algorithms from the Spark ML, (ii) Stage 2 is for misuse detection, which is based on the Conv-LSTM network. To deploy such an IDS in a real-life scenario, we further incorporate the alarm module. Overall, our IDS based on this two-stage learning system will be capable of more accurate anomaly and misuse detection.

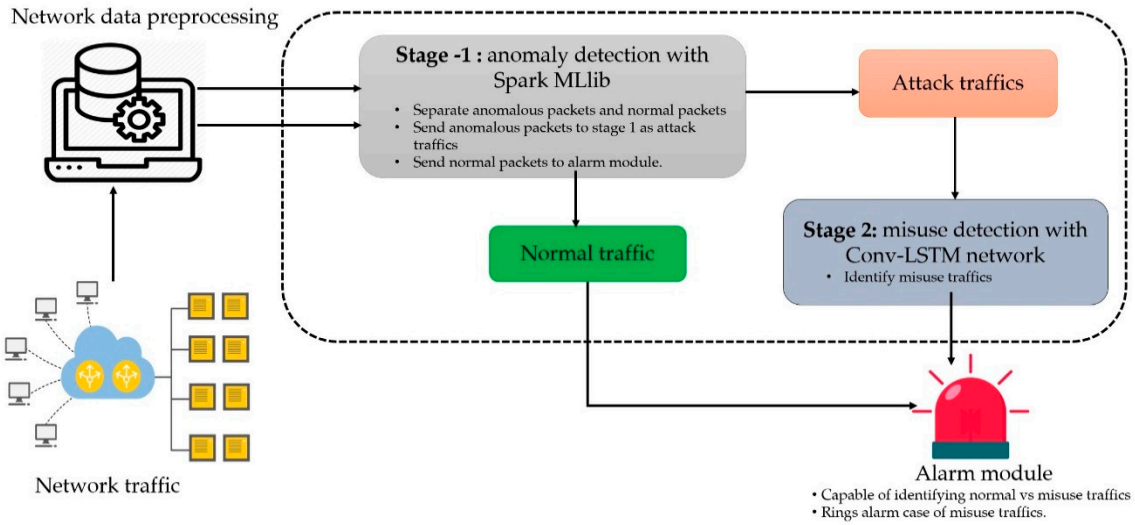

Figure 1. An overview of the proposed ID model.

3.2. Datasets

Since selecting the appropriate dataset to test a robust IDS system plays an important role, we describe and prepare the dataset before we discuss the implementation details of our proposed approach.

3.2.1. Description of the Dataset

Even though there are several benchmark ID datasets publicly available, several of them contain old-fashioned, undevitrified, inflexible, and irreproducible intrusions. To reduce these deficiencies and produce more contemporary traffic patterns, the ISCX-UNB dataset was produced by the Canadian Institute for Cybersecurity [46]. It includes several kinds of datasets to evaluate anomaly-based methods. The ISCX-IDS 2012 dataset shows realistic network behavior and comprises various intrusion

scenarios. Moreover, it is shared as an entire network capture with all interior traces to evaluate payloads for deep data packet analysis.

The ISCX-IDS 2012 ID dataset contains both normal and malicious network traffic activity of seven days. The dataset was produced by profiles including abstract representations of the actions and behaviors of traffic in the network. For example, communication between the source and destination host over HTTP protocol can be denoted by packets sent and received, termination point properties, and other analogous characteristics. This representation builds a single profile. These profiles create real network traffic for HTTP, SSH, SMTP, POP3, IMAP, and FTP protocols [46].

The ISCX-IDS 2012 includes two different profiles to create network traffic behavior and scenarios. The profile that originates the anomalous or multi-stage states of attacks is known as the α profile, while the β profile characterizes features and the mathematical dissemination of the process. For instance, the β profile can contain packet size distributions in the payload specific patterns, the request of time distribution of protocol, while the α profile is constructed depending on prior attack and contains sophisticated intrusions for the individual day. There are four attack scenarios in the entire dataset according to the α profile:

- Infiltrating the network from inside
- HTTP denial of service
- Distributed denial of service using an IRC botnet
- Brute force SSH.

The complete ISCX-IDS 2012 data are summarized in Table 2. As can be seen in Table 2, every attack scenario was applied for only a single day and two days contained only regular traffic. Also, the authors of [30] explain the diversity of the regular network behavior and the complexity of the attack scenarios.

Table 2. Summary of the ISCX-IDS 2012 dataset (daily traffic).

Days	Date	Description	Size (GB)
Friday	11 June 2010	Normal, hence no malicious activity	16.1
Saturday	12 June 2010	Infiltrating the network from inside and normal activity	4.22
Sunday	13 June 2010	Infiltrating the network from inside and normal activity	3.95
Monday	14 June 2010	HTTP denial of service and normal activity	6.85
Tuesday	15 June 2010	Distributed denial of service using an IRC Botnet	23.04
Wednesday	16 June 2010	Normal, hence no malicious activity	17.6
Thursday	17 June 2010	Brute force SSH and normal activity	12.3

3.2.2. Feature Engineering and Data Preparation

As shown in Figure 1, the dump from the network traffic was initially prepared and preprocessed. The ISCX ID 2012 dataset was analyzed; after preprocessing, data were collected over seven days with the practical and systematic conditions reflecting network packet traffic and intrusions. Explicitly, the dataset is labeled for regular and malicious flows for a total of 2,381,532, and 68,792 records in each respective class. The attacks observed on the original network traffic dataset were separated into two classes, normal and malicious/abnormal.

Additionally, a variety of multi-stage attack situations were performed to produce attack traces (e.g., infiltration from the inside, HTTP, DoS, DDoS via an Internet Relay Chat (IRC) botnet, and brute force Secure Shell (SSH)). The training and test dataset distributions utilized in this study are presented in Table 3. Sections 3.3.1 and 3.3.2 give further details.

Table 3. Distribution of ISCX-IDS 2012 training and testing dataset.

Dataset/Network Flows	#Feature	Training		Testing	
ISCX-UNB Saturday	8	85,222	1353	45,889	1353
ISCX-UNB Monday	8	108,945	2451	58,664	1320
ISCX-UNB Tuesday	8	347,308	24,295	187,012	13,083
ISCX-UNB Wednesday	8	339,470	0	182,793	0
ISCX-UNB Thursday	8	255,054	3381	137,338	1822

Note: For both set: left—Benign, right—Malicious.

The essential idea was to test the consistency of the proposed novel hybrid algorithm against unknown or anomaly attack via the misuse technique. Table 4 describes detail organizations of datasets for stage-2 Conv-LSTM (Misuse) classification level for training and testing the network.

Table 4. Distribution of the data for the second stage classifier.

Input	#Features	Attack Category
Training set	8	HTTP DoS, DDoS, and Botnet
Test set	8	Brute force SSH, HTTP DoS, DDoS, Botnet, and Brute force SSH

3.3. Implementation Details

Since the network traffic contains both malicious and normal traffic signatures, Spark ML-based classifiers are trained to categorize the data into malicious and normal classes in stage 1. However, the Conv-LSTM network-based stage deals with malicious traffic to achieve a higher degree of ID accuracy and a low false alarm rate (FAR).

3.3.1. Stage 1: The Anomaly Detection Module

In this stage, we used the Spark ML implementation of the SVM, DT, RF, and Gradient Boosting tree (GBT) classifiers to classify attack traffic (i.e., malicious versus normal traffic). We split the training set into two subsets: 80% for the training and 20% for the testing. The classifiers were trained on the training set to learn normal vs. malicious traffic, in a binary classification setting. Then the trained classifiers are evaluated on the test set. While training these algorithms, we performed 10-fold cross-validation and grid search for the hyperparameter tuning. In each case, the best performing model was selected to evaluate the test set.

3.3.2. Stage 2: Misuse Detection and Classification Module

In this module, Conv-LSTM, used for detecting misused attacks, aims to further categorize the malicious data from stages into corresponding classification strategies, i.e., Scan, R2L, DoS, and HTTP. In LSTM the DL-based misuse attack detection technique first trained the malicious traffic to generate a model that stated the baseline profile for malicious traffic only. A schematic representation of the Conv-LSTM network is shown in Figure 2. Intuitively, an end-to-end CNN has two components: (i) a feature extractor and (ii) a classifier. The feature extractor comprises two layers called convolution and pooling layers. The extracted output, which is known as the features map, becomes the input to the second component for the classification. In this way, CNN learns the local features very well. However, the downside is that it misses the long-range interdependency of important features. Therefore, to capture the local as well as the global features more robustly, we introduced LSTM layers [47–49] after the CNN layers. In this way, we managed to address the vanishing and exploding gradient problems efficiently, which enhances the ability to ensure longer dependencies and learn efficiently from variable extent sequences [50,51].

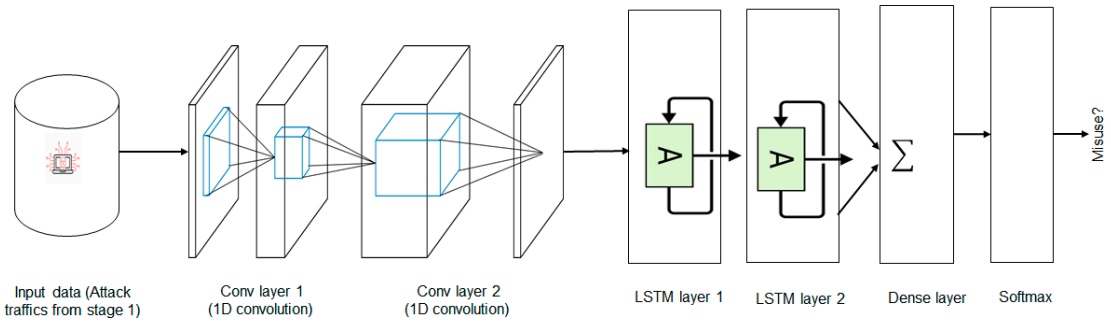

Figure 2. A schematic representation of the Conv-LSTM network, which starts by measuring attack, traffic and passing that data to both the CNN and LSTM layers before getting a flattened vector, which was fed through dense and Softmax layers for predicting the malicious traffic.

In the Conv-LSTM network, the input is initially processed by CNN, and then the output of CNN is passed through the LSTM layers to generate sequences at each time step, which helps us model both short-term and long-term temporal features [51]. Then the sequence vector is passed through a fully connected layer before feeding it into a Softmax layer for the probability distribution over the classes. In this stage, the test set is used as one of the inputs to the trained model to test if the behavior of trained traffic is normal or malicious. The attack traffic predicted by the classic models is also combined with the test set.

Then, similarly, we randomly split the dataset into training (80%) and test sets (20%) for testing. Also, 10% of the sample from the training set was used for the validation. During the training phase, first-order gradient-based optimization techniques such as Adam, AdaGrad, RMSprop, and AdaMax, with varying learning rates, were used to optimize the binary cross-entropy loss of the predicted network packet vs. the actual network packet, optimized with different combination of hyperparameters from grid search and 10-fold cross-validation to train each model on a batch size of 128. Also, we assessed the performance by adding Gaussian noise layers followed by Conv and LSTM layers to improve the model generalization and reduce overfitting.

3.4. The Alarm Module

When misuse is detected, the alarm module not only raises the alert but also compares it with normal traffic. The purpose of the alarm module is to interpret events' results on both the stage 1 and stage 2 modules. It is the last module of the proposed hybrid ID architecture that reports the ID activity to the administrator or end user.

4. Experimental Results

To show the effectiveness of our proposed hybrid approach on the ISCX ID 2012 ID dataset, we performed several experiments. We discuss the results both quantitatively and qualitatively.

4.1. Experimental Setup

The initial stage is implemented in Scala based on Spark ML. Conv-LSTM, on the other hand, was implemented in Python using Keras. Experiments were performed on a PC having a core i7 processor and 32 GB of RAM running 64-bit Ubuntu 14.04 OS. The software stack comprised of Apache Spark v2.3.0, Java (JDK) 1.8, Scala 2.11.8, and Keras. Eighty percent of the data was used for the training with 10-fold cross-validation and we evaluated the trained model based on the 20% held-over data. The Conv-LSTM is implemented in Keras and trained on an Nvidia TitanX GPU with CUDA and cuDNN, enabled to make the overall pipeline faster.

4.2. Performance Metrics

Once the models are trained, we evaluated them on the held-over test set. Then we computed the confusion matrix to compute the performance metrics. The elements of the confusion matrix help represent the predicted and expected/actual classification. The outcome of classifying is two classes: correct and incorrect. There are four fundamental situations that we considered to compute the confusion matrix:

- **True positive (TP)** measures the proportion of actual positives that are correctly identified. We specify this with x.
- **False negative (FN)** signifies the wrong predictions. More specifically, it identifies instances that are malicious but that the model incorrectly predicts as normal. We specify this with y.
- **False positive (FP)** signifies an incorrect prediction of positive, when in reality, the detected attack is normal. We specify this with z.
- **True negative (TN)** measures the proportion of actual negatives that are correctly identified attacks. We specify this with t.

Now, based on the above metrics x, y, z, and t, we have the confusion matrix in the intrusion detection setting as shown in Table 5.

Table 5. Confusion matrix for the IDS scheme.

	Predicted		
Actual	Normal	TP	FN
	Anomaly	FP	TN

From these conditions of the confusion matrix, we can calculate the performance of an IDS using the detection rate (DR) or true positive rate (TPR) and the false alarm rate (FAR), which are the two most fundamental general parameters for evaluating IDS. While DR or TPR means the ratio of intrusion instances identified by the ID model, FAR signifies the proportion of misclassified regular instances:

$$TPR = DR = TP/(TP + FN) = x/(x + y) \tag{1}$$

$$FAR = FP/(TN + FP) = z/(t + z). \tag{2}$$

When DR increases, FAR decreases. We then calculate our approach efficiency E and evaluate the hybrid IDS approach as follows:

$$E = DR/FAR. \tag{3}$$

4.3. Evaluation of the IDS System

Table 6 shows the performance of different classifiers at each stage. Only the results based on the best hyperparameters produced through an empirical random search are reported here. As shown in the table, classic model SVM performed quite poorly, giving an accuracy of only 68% in the F1-score. Tree-based classifiers managed to boost the performance significantly, showing accuracies of up to 89%.

Table 6. Performance of the classifiers at each stage.

Classifier	**Precision**	**Recall**	**F1-Score**	**FAR**	**DR**	**Stage**
SVM	0.6835	0.6515	0.6786	15.27	0.65	1
DT	0.7930	0.8012	0.7965	11.29	0.82	1
GBT	0.8529	0.8632	0.8612	8.13	0.85	1
RF	0.8919	0.8875	0.8845	5.72	0.89	1
Conv-LSTM	0.9725	0.9750	0.9729	0.71	0.97	2

However, the most significant boost that we experienced is with the Conv-LSTM network, which manages to accurately detect misuse in up to 97% of cases. The superior feature extraction of CNN and long-term dependencies between non-linear features is the reason behind this significant performance improvement. Implementation details are given in Supplementary Materials.

4.4. Overall Analysis

Table 7 compares our results with existing solutions for the ISCX-UNB dataset. This dataset was generated much later than the DARPA and KDD dataset family, so there are relatively fewer corresponding experimental results available [29]. Based on the available evaluation results for the compared methods, the best results for each study have been selected in relation to the accuracy and the false alarm rate.

Table 7. Comparison of our approach with existing solutions on the ISCX-UNB dataset.

Reference	Approach	Accuracy (DR)	False Alarm Rate
Kakavand et al. [25]	PCA	97.0	1.2
Kumar et al. [26]	AMGA2-NB	94.5	7.0
Tan et al. [29]	MCA + EMD	90.12	7.92
Sally et al. [30]	PLL + NGL	95.31	0.80
Our approach	Spark ML + Conv-LSTM	97.29	0.71

It can be observed that our proposed system performs better both in relation to the accuracy and the false alarm rate associated with advanced techniques, mainly because of the efficient feature selection technique we used and the implementation of a suitable Spark ML and Conv-LSTM approach. It is worth noting that these comparisons are for reference only as many researchers have used different proportions of traffic types and dataset distributions, preprocessing techniques, and sampling methods.

Therefore, a straightforward comparison of some metrics, such as training and testing time, is usually not considered appropriate, although our hybrid approach achieved improved performance in terms of all the performance metrics and outperformed other approaches. Nevertheless, we state that one can achieve a remarkable level of security against intrusion attacks using the hybrid technique, which is simple, fast, vigorous, and highly appropriate for real-time applications as well.

5. Conclusions and Outlook

In this paper, a hybrid IDS is proposed, implemented, and evaluated on the well-known ISCX-UNB dataset. The presented IDS is largely based on Spark ML and the Conv-LSTM network, which is particularly useful for the cybersecurity research. Our proposed IDS, which is a two-stage learning platform, shows good predictive accuracy at each stage, which is not only the case for the classification algorithms such as DT, RF, GBT, and SVM but also for the Conv-LSTM network, giving 97.29% predictive accuracy.

The proposed hybrid IDS based on the Conv-LSTM network integrated the power of both CNN and the LSTM network, which can be thought of as an efficient approach having both AB (Anomaly-based) and SB (Signature-based) classification approaches. The modular and hierarchical structure of our IDS not only performs better than state-of-the-art IDS approaches in terms of DR and accuracy for intrusion detection but also reduces the computational complexity.

However, one possible downside of our approach is that we have tested our IDS on only a single dataset. It is important to test it on a more recent dataset since the signature of the attack traffic often changes. In the future, we intend to: (i) extend our work so that anomaly and network misuses can be detected on real-time streaming data, and (ii) focus on exploring DL as an attribute extraction tool to learn competent data illustrations in case of other anomaly recognition problems in a more recent dataset.

Supplementary Materials: The source codes of the implementation are available on GitHub at https://github.com/rezacsedu/Intrusion-Detection-Spark-Deep-Learning.

Author Contributions: M.A.K. conceived the research, wrote the paper, designed the framework, and performed the experiments. M.R.K. developed the algorithm, helped with the experiments, and contributed to the background research. Y.K. and M.R.K. assisted with the proofreading, revision, and improvements. Y.K. supervised the overall research.

Funding: This research was supported by the MSIT (Ministry of Science, ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2018-2016-0-00465) supervised by the IITP (Institute for Information & communications Technology Promotion).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

RNN	Recurrent Neural Network
Conv-LSTM	Convolutional-Long short-term memory
IDS	Intrusion detection system
DL	Deep learning
ICT	Information and communication technology
CIA	Confidentiality, integrity, and availability
SBS	Signature-based system
HIDS	Host intrusion detection system
NIDS	Network intrusion detection system
DoS	Denial of service
U2R	User to root
R2L	Remote to local STL
STL	Self-taught learning
NN	Neural network
MLP	Multilayer perceptron
SVM	Support vector machine
GBT	Gradient Boosting tree
DBN	Deep belief network
DSS	Decision support system
DT	Decision tree
SSO	Simplified swarm optimization
NB	Naive Bayes
ESVDF	Enhanced Support Vector Decision Function

References

1. Xu, C.; Shen, J.; Du, X.; Zhang, F. An intrusion detection system using a deep neural network with gated recurrent units. *IEEE Access* **2018**, *6*, 48697–48707. [CrossRef]
2. Vinayakumar, R.; Soman, K.P.; Poornachandran, P. Applying convolutional neural network for network intrusion detection. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017; pp. 1222–1228.
3. Sharma, S.; Gupta, R. Intrusion detection system: A review. *Int. J. Secur. Its Appl.* **2015**, *9*, 69–76. [CrossRef]
4. Allen, J.; Christie, A.; Fithen, W.; Mchugh, J.; Pickel, J. *State of the Practice of Intrusion Detection Technologies*; Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst: Pittsburgh, PA, USA, 2000.
5. Mighan, S.N.; Kahani, M. Deep Learning Based Latent Feature Extraction for Intrusion Detection. In Proceedings of the Iranian Conference on Electrical Engineering (ICEE), Mashhad, Iran, 8–10 May 2018; pp. 1511–1516.
6. Bijone, M. A survey on secure network: Intrusion detection prevention approaches. *Am. J. Inf. Syst.* **2016**, *4*, 69–88.
7. Hodo, E.; Bellekens, X.; Hamilton, A.; Tachtatzis, C.; Atkinson, R. Shallow and deep networks intrusion detection system: A taxonomy and survey. *arXiv* **2017**, arXiv:1701.02145.

8. Axelsson, S. *Intrusion Detection Systems: A Survey and Taxonomy*; Technical Report; Chalmers University: Goteborg, Sweden, 2000; Volume 99.
9. Kim, J.; Kim, H. An effective intrusion detection classifier using long short-term memory with gradient descent optimization. In Proceedings of the IIEEE international Conference on Platform Technology and Service (PlatCon), Busan, Korea, 13–15 February 2017.
10. Hinton, G.E.; Osindero, S.; Teh, Y.W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef]
11. Wu, Z.; Wang, X.; Jiang, Y.G.; Ye, H.; Xue, X. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In Proceedings of the 23rd ACM International Conference on Multimedia, Brisbane, Australia, 26–30 October 2015; pp. 461–470.
12. Tang, D.; Qin, B.; Liu, T. Document modeling with gated recurrent neural network for sentiment classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Lisbon, Portugal, 17–21 September 2015; pp. 1422–1432.
13. Fan, Y.; Lu, X.; Li, D.; Liu, Y. Video-based emotion recognition using CNN-RNN and C3D hybrid networks. In Proceedings of the 18th ACM International Conference on Multimodal Interaction, Tokyo, Japan, 12–16 November 2016; pp. 445–450.
14. Vignesh, K.; Yadav, G.; Sethi, A. Abnormal Event Detection on BMTT-PETS Surveillance Challenge. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 2161–2168.
15. Yin, C.; Zhu, Y.; Fei, J.; He, X. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access* **2017**, *5*, 21954–21961. [CrossRef]
16. Kuang, F.; Xu, W.; Zhang, S. A novel hybrid KPCA and SVM with GA model for intrusion detection. *Appl. Soft Comput.* **2014**, *18*, 178–184. [CrossRef]
17. Reddy, R.R.; Ramadevi, Y.; Sunitha, K.V.N. Effective discriminant function for intrusion detection using SVM. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, India, 21–24 Septembert 2016; pp. 1148–1153.
18. Li, W.; Yi, P.; Wu, Y.; Pan, L.; Li, J. A new intrusion detection system based on KNN classification algorithm in wireless sensor network. *J. Electr. Comput. Eng.* **2014**, *2014*, 240217. [CrossRef]
19. Farnaaz, N.; Jabbar, M.A. Random forest modeling for network intrusion detection system. *Procedia Comput. Sci.* **2016**, *89*, 213–217. [CrossRef]
20. Zhang, J.; Zulkernine, M.; Haque, A. Random-Forests-Based Network Intrusion Detection Systems. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2008**, *38*, 649–659. [CrossRef]
21. Kim, G.; Lee, S.; Kim, S. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Syst. Appl.* **2014**, *41*, 1690–1700. [CrossRef]
22. Panda, M.; Patra, M.R. Network intrusion detection using naive bays. *Int. J. Comput. Sci. Netw. Secur.* **2007**, *7*, 258–263.
23. Zaman, S.; Karray, F. Features selection for intrusion detection systems based on support vector machines. In Proceedings of the IEEE Consumer Communications and Networking Conference, Las Vegas, NV, USA, 10–13 January 2009; pp. 1–8.
24. Depren, O.; Topallar, M.; Anarim, E.; Ciliz, M.K. An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. *Expert Syst. Appl.* **2005**, *4*, 713–722. [CrossRef]
25. Kakavand, M.; Mustapha, N.; Mustapha, A.; Abdullah, M.T. Effective Dimensionality Reduction of Payload-Based Anomaly Detection in TMAD Model for HTTP Payload. *KSII Trans. Internet Inf. Syst.* **2016**, *10*, 3884–3910.
26. Kumar, G.; Kumar, K. Design of an evolutionary approach for intrusion detection. *Sci. World J.* **2013**, *2013*, 962185. [CrossRef]
27. Yassin, W.; Udzir, N.I.; Muda, Z.; Sulaiman, M.N. Anomaly-based intrusion detection through k-means clustering and naives Bayes classification. In Proceedings of the 4th International Conference on Computing and Informatics, ICOCI, Kuching, Malaysia, 28–30 August 2013; Volume 49, pp. 298–303.
28. Tahir, H.M.; Said, A.M.; Osman, N.H.; Zakaria, N.H.; Sabri, P.N.A.M.; Katuk, N. Oving K-means clustering using discretization technique in network intrusion detection system. In Proceedings of the 3rd International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, Malaysia, 15–17 August 2016; pp. 248–252.

29. Tan, Z.; Jamdagni, A.; He, X.; Nanda, P.; Liu, R.P.; Hu, J. Detection of Denial-of-Service Attacks Based on Computer Vision Techniques. *IEEE Trans. Comput.* **2015**, *64*, 2519–2533. [CrossRef]
30. Sallay, H.; Ammar, A.; Saad, M.B.; Bourouis, S. A real time adaptive intrusion detection alert classifier for high speed networks. In Proceedings of the IEEE 12th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 22–24 August 2013; pp. 73–80.
31. Ingre, B.; Yadav, A. Performance analysis of NSL-KDD dataset using ANN. In Proceedings of the IEEE International Conference on Signal Processing and Communication Engineering Systems, Guntur, India, 2–3 January 2015; pp. 92–96.
32. Lipton, Z.C.; Berkowitz, J.; Elkan, C. A critical review of recurrent neural networks for sequence learning. *arXiv* **2015**, arXiv:1506.00019.
33. Gao, N.; Gao, L.; Gao, Q.; Wang, H. An intrusion detection model based on deep belief networks. In Proceedings of the IEEE Second International Conference on Advanced Cloud and Big Data, Huangshan, China, 20–22 November 2014; pp. 247–252.
34. Moradi, M.; Zulkernine, M. A neural network-based system for intrusion detection and classification of attacks. In Proceedings of the IEEE International Conference on Advances in Intelligent Systems-Theory and Applications, Guwahati, India, 4–6 March 2004; pp. 15–18.
35. Staudemeyer, R.C.; Omlin, C.W. Extracting salient features for network intrusion detection using machine learning methods. *S. Afr. Comput. J.* **2014**, *52*, 82–96. [CrossRef]
36. Staudemeyer, R.C.; Omlin, C.W. Evaluating performance of long short-term memory recurrent neural networks on intrusion detection data. In Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference, East London, Africa, 7–9 October 2013; pp. 218–224.
37. Staudemeyer, R.C. Applying long short-term memory recurrent neural networks to intrusion detection. *S. Afr. Comput. J.* **2015**, *56*, 136–154. [CrossRef]
38. Mukkamala, S.; Sung, A.H.; Abraham, A. Intrusion detection using an ensemble of intelligent paradigms. *J. Netw. Comput. Appl.* **2005**, *28*, 167–182. [CrossRef]
39. Javaid, A.; Niyaz, Q.; Sun, W.; Alam, M. A deep learning approach for network intrusion detection system. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), New York, NY, USA, 3–5 December 2016; pp. 21–26.
40. Faraoun, K.M.; Boukelif, A. Neural networks learning improvement using the K-means clustering algorithm to detect network intrusions. *INFOCOMP* **2006**, *5*, 28–36.
41. Santos, L.; Rabadao, C.; Gonçalves, R. Intrusion detection systems in Internet of Things: A literature review. In Proceedings of the IEEE 13th Iberian Conference on Information Systems and Technologies (CISTI), Caceres, Spain, 13–16 June 2018; pp. 1–7.
42. Tsiropoulou, E.E.; Baras, J.S.; Papavassiliou, S.; Qu, G. On the Mitigation of Interference Imposed by Intruders in Passive RFID Networks. In *International Conference on Decision and Game Theory for Security*; Springer: Cham, Switzerland, 2016; pp. 62–80.
43. Ghanem, T.F.; Elkilani, W.S.; Abdul-Kader, H.M. A hybrid approach for efficient anomaly detection using metaheuristic methods. *J. Adv. Res.* **2015**, *6*, 609–619. [CrossRef]
44. Sabhnani, M.; Serpen, G. Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context. In Proceedings of the International Conference on Machine Learning: Models, Technologies, and Applications (MLMTA), Las Vegas, NV, USA, 23–26 June 2003; pp. 209–215.
45. Chung, Y.Y.; Wahid, N. A hybrid network intrusion detection system using simplified swarm optimization (SSO). *Appl. Soft Comput.* **2012**, *12*, 3014–3022. [CrossRef]
46. Shiravi, A.; Shiravi, H.; Tavallaee, M.; Ghorbani, A.A. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Comput. Secur.* **2012**, *31*, 357–374. [CrossRef]
47. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
48. Gers, F.A.; Schraudolph, N.N.; Schmidhuber, J. Learning precise timing with LSTM recurrent networks. *J. Mach. Learn. Res.* **2002**, *3*, 115–143.
49. Giancarlo, Z.; Karim, M.R. *Deep Learning with TensorFlow: Explore Neural Networks and Build Intelligent Systems with Python*; Packt Publishing Ltd.: Birmingham, UK, 2018.

50. Khan, M.A.; Karim, M.R.; Kim, Y. A Two-Stage Big Data Analytics Framework with Real World Applications Using Spark Machine Learning and Long Short-Term Memory Network. *Symmetry* **2018**, *10*, 485. [CrossRef]
51. Karim, M.R.; Cochez, M.; Dietrich-Rebholz, S. Recurrent Deep Embedding Networks for Genotype Clustering and Ethnicity Prediction. *arXiv* **2018**, arXiv:1805.12218.