ï»¿ In Network and Distributed Systems Security Sym posium (N DS S) 2018 , San Diego , February 2018 Kits une : An En semble of Aut oen cod ers for Online Network Int r usion Detection Y is ro el Mir sky , Tom er Do its h man , Yu val El ovic i and As af Sh ab t ai ar X iv : 18 02 . 09 089 v 1 [cs . CR] 25 Feb 2018 Ben - G urion University of the Ne ge v { y is ro el , to mer doi } @ post . b gu . ac . il , { el ovic i , sh ab ta ia } @ b gu . ac . il Abstract Ne ural networks have become an increasingly popular solution for network intrusion detection systems (N IDS). Their capability of learning complex patterns and behaviors make them a suitable solution for different iating between normal traffic and network attacks . However , a drawback of neural networks is the amount of resources needed to train them . Many network gate ways and routers devices , which could potentially host an N IDS , simply do not have the memory or processing power to train and sometimes even execute such models . More importantly , the existing neural network solutions are trained in a supervised manner . Meaning that an expert must label the network traffic and update the model manually from time to time . In this paper , we present Kits une : a plug and play N IDS which can learn to detect attacks on the local network , without supervision , and in an efficient online manner . Kits une s core algorithm (Kit NET) uses an ensemble of neural networks called aut oen cod ers to collectively differentiate between normal and abnormal traffic patterns . Kit NET is supported by a feature extraction framework which efficiently tracks the patterns of every network channel . Our evaluations show that Kits une can detect various attacks with a performance comparable to offline anomaly detectors , even on a Raspberry PI . This demonstrates that Kits une can be a practical and economic N IDS . Key words An omaly detection , network intrusion detection , online algorithms , aut oen cod ers , ensemble learning . I . I N TR ODUCT ION The number of attacks on computer networks has been increasing over the years [1]. A common security system used to secure networks is a network intrusion detection system (N IDS). An N IDS is a device or software which monitors all traffic passing a strategic point for malicious activities . When such an activity is detected , an alert is generated , and sent to the administrator . Convention ally an N IDS is deployed at a single point , for example , at the Internet gateway . This point deployment strategy can detect malicious traffic entering and leaving the network , but not malicious traffic travers ing the network itself . To resolve this issue , a distributed deployment strategy can be used , where a number of N IDS s are be connected to a set of strategic routers and gate ways within the network . Per mission to freely reproduce all or part of this paper for non commercial purposes is granted provided that copies bear this notice and the full citation on the first page . Reprodu ction for commercial purposes is strictly prohibited without the prior written consent of the Internet Society , the first - named author (for reproduction of an entire paper only), and the author s employer if the paper was prepared within the scope of employment . N DS S 18 , 18 - 21 February 2018 , San Diego , CA , USA Copyright 2018 Internet Society , ISBN 1 - 189 15 62 - 49 - 5 http :// dx . doi . org / 10 . 147 22 / nd ss . 2018 . 23 204 Over the last decade many machine learning techniques have been proposed to improve detection performance [2], [3], [4]. One popular approach is to use an artificial neural network (ANN) to perform the network traffic inspection . The benefit of using an ANN is that ANN s are good at learning complex non - linear concepts in the input data . This gives ANN s a great advantage in detection performance with respect to other machine learning algorithms [5], [2]. The prevalent approach to using an ANN as an N IDS is to train it to classify network traffic as being either normal or some class of attack [6], [7], [8]. The following shows the typical approach to using an ANN - based class ifier in a point deployment strategy : 1) Have an expert collect a dataset containing both normal traffic and network attacks . 2) Train the ANN to classify the difference between normal and attack traffic , using a strong CPU or GPU . 3) Transfer a copy of the trained model to the network / organ ization s N IDS . 4) Have the N IDS execute the trained model on the observed network traffic . In general , a distributed deployment strategy is only practical if the number of N IDS s can economically scale according to the size of the network . One approach to achieve this goal is to embed the N IDS s directly into inexpensive routers (i . e ., with simple hardware). We argue that it is impractical to use ANN - based class ifiers with this approach for several reasons : Offline Processing . In order to train a supervised model , all labeled instances must be available locally . This is inf eas ible on a simple network gateway since a single hour of traffic may contain millions of packets . Some works propose off loading the data to a remote server for model training [9] [3]. However , this solution may incur significant network overhead , and does not scale . Super vised Learning . The labeling process takes time and is expensive . More importantly , what is considered to be normal depends on the local traffic observed by the N IDS . Furthermore , in attacks change overtime and while new ones are constantly being discovered [10], so continuous maintain able of a malicious attack traffic repository may be impractical . Finally , classification is a closed - world approach to identifying concepts . In other words , a class ifier is trained to identify the classes provided in the training set . However , it is unreasonable to assume that all possible classes of malicious traffic can be collected and placed in the training data . High Complex ity . The computational complexity of an ANN Output Layer En semble Layer Map score RM SE RM SE RM SE RM SE RM SE RM SE RM SE RM SE The reason we use aut oen cod ers is because (1) they can trained in an un super vised manner , and (2) they can be used for anomaly detection in the event of a poor reconstruction . The reason we propose using an ensemble of small aut oen cod ers , is because they are more efficient and can be less no is ier than a single aut oen c oder over the same feature space . From our experiments , we found that Kits une can increase the packet processing rate by a factor of five , and provide a detection performance which rivals other an offline (batch) anomaly detectors . In summary , the contributions of this paper as follows : A novel aut oen c oder - based N IDS for simple network devices (K its une), which is lightweight and plug - and - play . To the best of our knowledge , we are the first to propose the use of aut oen cod ers with or without en semb les for online anomaly detection in computer networks . We also present the core algorithm (Kit NET) as a generic online un super vised anomaly detection algorithm , and provide the source code for download . 1 A feature extraction framework for dynamically maintaining and extracting implicit contextual features from network traffic . The framework has a small memory footprint since the statistics are updated increment ally over d amped windows . An online technique for automatically constructing the ensemble of aut oen cod ers (i . e ., mapping features to ANN inputs) in an un super vised manner . The method involves the incremental hierarch al clust ering of the feature - space (trans pose of the unb ounded dataset), and bound ing of cluster sizes . Experimental results on an operational IP camera video surveillance network , IoT network , and a wide variety of attacks . We also demonstrate the algorithm s efficiency , and ability to run on a simple router , by performing benchmarks on a Raspberry PI . Fig . 1 : An illustration of Kits une s anomaly detection algorithm Kit NET . grows exponentially with number of neurons [11]. This means that an ANN which is deployed on a simple network gateway , is restricted in terms of its architecture and number of input features which it can use . This is especially problematic on gate ways which handle high velocity traffic . In light of the challenges listed above , we suggest that the development of an ANN - based network intrusion detector , which is to be deployed and trained on routers in a distributed manner , should adhere to the following restrictions : Online Processing . After the training or executing the model with an instance , the instance is immediately discarded . In practice , a small number of instances can be stored at any given time , as done in stream clust ering [12]. Un super vised Learning . Lab els , which indicate explicitly whether a packet is malicious or benign , are not used in the training process . Other meta information can be used so long as acquiring the information does not delay the process . Low Complex ity . The packet processing rate must exceed the expected maximum packet arrival rate . In other words , we must ensure that there is no queue of packets awaiting to be processed by the model . The rest of the paper is organized as follows : Section II discusses related work in the domain of online anomaly detection . Section III provide a background on aut oen cod ers and how they work . Section IV presents Kits une s framework and it s entire machine learning pipeline . Section V presents experimental results in terms of detection performance and run - time performance . Finally , in section VII we present our conclusion . In this paper , we present Kits une : a novel ANN - based N IDS which is online , un super vised , and efficient . A Kits une , in Japanese folklore , is a mythical fox - like creature that has a number of tails , can mimic different forms , and whose strength increases with experience . Similarly , Kits une has an ensemble of small neural networks (aut oen cod ers), which are trained to mimic (re construct) network traffic patterns , and whose performance increment ally improves overtime . ï»¿