Survey of Applied Machine Learning in Computer Network Communication and Management

Dafydd Foster
Department of Computer Science
Dafyddfoster@gmail.com

Jacob Thom
Department of Computer Science
Thejacobthom@gmail.com

Abstract

In an ever-changing world with the emergence of zero-day threats becoming ever more common, it is imperative that the networks of the world possess the capabilities to defend and protect themselves from malicious attacks without active human involvement and oversight. It is likewise of great importance that the networks of the world possess the ability to adapt to the ever-increasing internet usage found in the modern world, without surges of activity causing congestion, delay, or loss of data. With unforeseen situations causing conflict and inducing stress upon worldwide networks, such as during the coronavirus pandemic [1], a solution needs to be implemented that can respond to and circumvent disaster without human involvement. With the widespread use of algorithms, actively maintained databases, and human involvement, currently adopted solutions are able to reduce conflict for the most part. It goes without saying that there is clear room for improvement, such as may be seen in the reduction of human involvement. In this paper we review machine learning applications and their ability to improve upon existing solutions to the issues discussed above through careful application.

1. Introduction

This paper sets out to fulfill what are believed to be two areas of concern in existing surveys concerning the applications of machine learning. The first concern is shown in the presentation of both [2], [3], and [4] which while being recent and encompassing their specific area of research, do not serve to present a broadened picture for the applications of machine learning. The second concern is seen in that the most recent survey for an encompassing picture of machine learning applications was completed and published in 2019 [5]. During the time since the encompassing survey was completed, it is the belief of our team that it has become outdated, with areas of application mentioned within being further researched and developed. With these two concerns in mind, this survey sets out to fulfill both of them and therefore present a modern picture for the applications of machine learning on computer networks in 2021.

In this survey we have divided the contents into three parts which are believed to be the cornerstones in the applications for machine learning. The first and most important application is addressed in *Section 3* wherein the current research progress and applications of traffic classification are both presented and discussed. This is perceived to be the area of greatest importance in machine learning applications, as all other applications utilize traffic classification and observation in order to fulfill their individual mandates. In *Section 4*, the applications of machine learning concerning mitigation of network threats such denial-of-service attacks (DOS) and distributed denial-of-service attacks (DDOS) are discussed and reviewed. *Section 5* discusses the usage of machine learning for improved congestion control for network traffic. In *Section 2* the concepts surrounding the implementation and application of machine learning are briefly introduced.

2. Background Information

Machine learning is a technique in computer science that focuses on the development of novel algorithms under three main branches of implementation of algorithmic learning: supervised, unsupervised, and reinforcement. Machine learning is a valuable tool, when properly implemented it is able to outperform humans in areas such as problem recognition, reaction time, and solution development. These traits are desirable in a world that is developing faster than any single team or individual can comprehend, machine learning allows massive datasets to be aggregated into coherent results that previously may have been foregone by human operators or taken longer to be realized. This ability to coalesce multiple data points into actionable values and methods is why machine learning is well suited to tackle problems facing computer networks during modern times. Machine learning is aptly suited to both aid in development and for integration into computer networks so that they may possess the ability to operate in a hands-off, optimized approach that is able to better serve a computer network than algorithms that currently exist for traffic observation, threat mitigation, and traffic congestion control.

Machine learning operates under the three branches that will be briefly discussed in this section before being showcased in their use-cases in the following sections. Supervised learning operates using labelled data and seeks to predict a determined outcome/future event based upon the data given. Unsupervised learning utilizes non-labelled data and is used to locate and classify data structures (such as patterns) that exist within the dataset to develop a better understanding of the data and allow for segregation into various structures. Reinforcement learning is a form of machine learning that focuses on developing an algorithm by using a reward structure to encourage ‘good’ results (such as predictions, classifications, patterns, and outcomes) and discourage unfavourable results through punishment (reduction or lack of a reward). These algorithmic methods all work by receiving substantially large enough sets of data so that they may develop and/or utilize internal algorithms to coalesce the information into clear, observable output. In the context of this survey and this domain of research, the output may then be used in a decision-making process such as whether or not to block an incoming network connection, relay information to a node in the network, or incentivize a change in software defined networking (SDN) parameters.

3. Traffic Classification

Accurately classifying network traffic is one of the most crucial tasks to be performed. Unfortunately, due to the contextual nature of what is considered an accurate classification, this is also one of the hardest tasks. The dimensionality of the classification plane may range from a binomial distribution, in regard to threat detection in its simplest form, to a theoretically boundless range in regards to network routing. Traditional methods of traffic classification involve inspecting either the port or payload to determine functionality. Although effective, techniques of this variety introduce extra round trip time due to the need to read each packet which traverses the network. This introduction of packet delay is a focus in which machine learning techniques are trying to resolve.

3.1 Current Progress

As the name suggests, this is solely a classification problem, and thus, although unsupervised learning techniques may be utilized in the data pre-processing, supervised learning is most used for the final classification. Due to the nonlinearity seen in network traffic data, all work reviewed throughout this paper chose to utilize forms of neural networks for classification. Although not explicitly mentioned, it is suspected tree-based classification methods suffer from high error rates due to the complexity of the data.

One of the issues when attempting to classify network traffic is the large dimensionality. Fundamentally, this is an issue due to the fact that often, not always, only a small subset of features in the dataset are utilized in accurately classifying the sample in question. The features which are not needed in this process create data noise, and potentially introduce patterns which will detrimentally skew the classification taking place. To help combat this issue, dimension reduction methods are often used. [6] does analysis on three distinct dimension reduction methods: Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), and Autoencoding (AE).

PCA selects features by their variance, with the goal of maintaining as much variance as possible. This selection is accomplished by utilizing a subset of orthogonal components from the dataset. Then, the linear combination which yields the highest variance from this subset is determined. Resulting from the linear combination, high dimensional data can be projected onto lower dimensions.

t-SNE initializes by calculating the pairwise distance between all samples. Following, all samples are randomly projected onto the desired dimension. The stochastic characteristics are used during placement refinement. On each step, pairs which were close in the original dimension are adjusted such that their distance in the new dimension is minimized. Likewise, pairs which were far apart in the original dimension are adjusted such that their distance in the new dimension is maximized. Due to the stochastic nature of t-SNE, its runtime is often much higher than other methods mentioned.

AE is a special classification of neural networks. These differ significantly in comparison to other neural networks as, unlike traditional neural networks which have a tree structure shape, AEs are structured as an hourglass. This is due to the fact that the wanted output does not get produced by the last layer, but from the center layer. The purpose of an AE is to find a way to compress, or re-represent, the given data through the use of a lossless function. Therefore, it is possible to reproduce the original data from the compressed version. The first half of an AE represents the encoder, which is in charge of determining a suitable lossless function. The second portion represents the decoder, which attempts to reproduce the data upon compression. Once training is completed, the AE is split into the two distinct parts, where the encoder portion is retained for the purpose of dimension reduction.

For each dimensional reduction method, [6] did analysis on both a semi-supervised and unsupervised model, OCSVM and DBSCAN respectively. The dataset used for testing consisted of 31 dimensions. Interestingly, the resulting scores showed in all cases in which dimensional reduction of any degree was used in conjunction with OCSVM, severe decreases to the accuracy occured. Contrastingly, utilizing DBSCAN, better performance was achieved compared to using the full feature set. In particular, the combination of DBSCAN with t-SNE reduction depicted a FP rate reduction of 16%, while utilizing only 2 of the 31 dimensions. This is substantial, not only due to the improved accuracy, but decrease in model complexity, and therefore, operation times.

As can be seen in the papers below, dimensional reduction is a key component of many protocols. This reduction is largely seen in the pre-processing of data as well as the processing of incoming data such that it fits the needs and comparisons drawn within a protocol. An example of this may be seen in [7] which utilizes density based spatial clustering of application of noise (DBSCAN) to determine data clusters for statistical feature extraction, allowing for an algorithm that focuses on key differentiating factors rather than those that may be inconsequential (and use unnecessary processing power).This is not to say that all algorithms attempt dimensionality reduction but seeks to highlight the fact that it is used to increase algorithm performance in traffic classification.

3.2 Future Work

It is no surprise that many advancements have been seen in the area of traffic classification since, no matter the application, if machine learning is to be utilized in networks, the first step needed is to correctly and accurately classify data. [6] demonstrated compelling results for the use of dimensional reduction alongside unsupervised models for physical layer anomaly classification. [7] saw slight accuracy improvement though the utilization of a multi machine learning model method in comparison to historical port-based techniques, although runtime analysis was not completed. To properly determine if their proposed method is superior, future research will need to take place with the inclusion of a more encompassing success metric plane.

4. Threat Mitigation

Threat recognition at current standards widely utilizes actively updated and maintained databases of threats as they come to light and are realized by the research community. This method is highly effective against threats that are known, but in the instance of zero-day threats these preventative measures fail miserably. Another method widely used, such as that by Solar Winds in [9] is to allow network administrators the ability to monitor and verify a non-threatened network and use this network as a baseline. Under [9], deviations from the baseline may have pre-chosen responses such as blocked connections or throttled network access. While these systems may be effective at either denying all known threats or denying all abnormal network access, they lack the flexibility required in modern networks and the security measures to deal with zero-day threats. This gap in application abilities is perfect for machine learning implementations. As will be seen in this section, machine learning presents the unique ability of actively monitoring a system, recognizing threats in said system, and handling them accordingly such that they do not impair or impart negative effects upon a network or connection.

4.1 Current Progress

At present due to the open space in which zero-day network threats may emerge, it is required by machine learning based protocols that they possess the ability to adapt to new and ever changing networks. The ability to adapt to new network conditions is paramount in a trained algorithm's ability to filter and prevent threats without impeding benign network traffic. As such, emerging protocols that seek to utilize machine learning largely focus on techniques with the ability to integrate incoming data into their algorithms. These algorithms at present largely focus on supervised learning and constitute a majority of this section. To provide an accurate depiction of the field, research forays into unsupervised learning and reinforcement learning will also be discussed.

Eric Perruad developed an unsupervised learning algorithm [10] that utilizes a developed k-means algorithm. Initially the algorithm will observe the network for a determined number of steps, these observations are then normalized and then transposed onto subspaces from which k-means and feature extraction are applied to gather the necessary statistics for an abnormality equation. Once this subspace and equation are developed the incoming packets are (once normalized) run through the abnormality equation. A significant enough abnormality deems the packet as a threat and sicards it while a non-threatening packet is otherwise integrated into the existing subspace and derived equation. Under Perraud's observation this technique caught all attempted DOS attack packets while maintaining a "very good false alarm rate".

[11] combines both supervised and unsupervised learning methods in their proposed Deep Auto-Encoder Intrusion Detection System (DAE-IDS). Deep auto-encoders are multiple auto-encoders joined together in a daisy-chain fashion. The proposed DAE consisted of four encoders. [11] mentioned one reason for using auto-encoders was to utilize the dimensional reduction characteristics in which they bring. The supervised learning aspect is introduced at the final layer, where a supervised model attempts to classify the data that was outputed from the DAE. To help mitigate overfitting issues, a greedy layer-wise training approach was used. Greedy layer-wise training is a scheme where each layer is sequentially trained in a bottom up manner. This technique has been shown to increase test accuracy in deep neural networks, due to an increased generalization of the model, although higher training times are introduced. Testing took place through the use of the KDD_CUP'99 dataset. Through analysis, it was shown the optimal number of hidden layers and corresponding neurons was 4 hidden layers, with 32 neurons for each layer, resulting in an accuracy of 94.71%, with a 94.53% detection rate. To evaluate the proposed DAE-IDS's competitiveness in the field it was compared to two prior existing models, DBN[4] and AutoEncoder+DBN[10-10]. DBN[4] and AutoEncoder+DBN[10-10] received an accuracy score of 93.49% and 92.10% respectively. Unfortunately, the detection rate for these pre-existing methods were not recorded,

therefore, although DAE-IDS showed higher accuracy then its predecessors, an inclusive comparison is not possible.

[12] analyses the performance of using Convolutional Neural Networks, CNN, for anomaly detection. CNN's are popular models where complex data is needed, with their most prominent use being with pictures. [12] tested three distinct models on three distinct datasets. The models used were shallow, moderate, and deep, where the naming represents the amount of convolutional and pooling layers used. The three datasets used for testing were NSL-KDD, Kyoto Honeypot, and MAWILab. Kyoto Honeypot is an unbalanced dataset while the other two are both balanced, where balanced and unbalanced refer to the ratio of normal traffic and attack traffic seen. Tests showed that either the shallow model outperformed both the moderate and deep model, in regards to NSL-KDD dataset, or there was no noticeable higher performance achieved between all three models, in relation to Kyoto Honeypot and MAWILab. Upon comparing results to other NN techniques it was found CNN's, on average, under performed. [12] hypothesizes this is due to the 1 dimensional vector which represents network traffic as CNN's are most commonly used in conjunction with 2 dimensional matrices.

[13] proposes a Deep Belief Neural Network, DBN. Testing on the NSL-KDD dataset, which was previously seen in [12], resulted in a detection accuracy of 97.5%, a 4.66% increase over existing DBN-SVM models which the proposed DBN was compared against. Although important, detection accuracy is not the only crucial metric when the

goal is an optimal real time intrusion detection system, execution time is just as important. The proposed DBN achieved its accuracy in 0.32 seconds, a substantial decrease in comparison to DBN-SVM's seen time of 3.07 seconds.

Unlike most papers reviewed thus far, which primarily focus on intrusion detection, [14] proposes a Autonomous Threat Mitigation framework, ATMoS. The goal of ATMoS is to provide network administration a platform for rapid design and deployment of reinforcement learning agents and consists of three parts: SDN infrastructure, host behaviour profiling, and autonomous management. The SDN infrastructure holds two key components: controller, and observer. As the naming suggests, the controller's job is to act upon the commands outputted by the agent, while the observer provides the agent with stateful information about the network. Host behaviour profiling is applied during training time, enabling the agent to learn normal network behaviour along with malicious actions. Autonomous Management is where the agent resides, along with where all ML decisions are made. [14] determined the success of ATMoS could be represented as the ratio between benign users' quality of experience and attack success. It is hypothesised that this definition of success will result in a more generalized model which, instead of classifying distinct attacks, allows for the classification of attacks based on their class. If correct, this allows for the possibility of the agent being able to accurately detect and mitigate attacks which were previously unseen as long as their class is known. ATMoS utilizes NFQ for its reinforcement learning algorithm. A recurring issue with RL in the field of intrusion mitigation is in designing a realistic action space for the agent. [14] decided to utilize virtual networks, VN, to represent differing security levels. Therefore, the action space can be represented with three actions: move a host up a security level VN, down a security level VN, or keep the host at the same security level VN. [14] represents a proof of concept of their proposed ATMoS, and thus rigorous testing did not occur. Through simulations It was shown ATMoS successfully detected and then mitigated both a TCP SYN-flood attack along with a Advanced Persistent Threat, APT.

4.2 Future Work

Given the uniformity under which threat mitigation may be tested, future work may be focused on analyzing all individual algorithms under the conditions proposed by a single dataset. This coalescing of analysis onto a standard testbed would provide a solid base for comparing, contrasting, and improving upon future algorithms. Similarly, common performance metrics need to be agreed upon to realistically compare differing implementations. [12] suggests further research is needed to see if network traffic can be efficiently mapped into a two dimensional structure, with the hope of increasing accuracy seen by CNN's in regards to anomaly detection. [13] showed interests in comparing existing results against a State Preserving Extreme Learning Machine, SPEML, implementation. [14] showed success through their proof of concept simulations, although more inclusive testing is required to determine the efficiency of the proposed framework.

5. Traffic Congestion Control

Traffic congestion in a simple network is primarily monitored and controlled for by individual nodes in modern contexts that do not utilize machine learning. What machine learning provides towards congestion control is an increased understanding of network congestion particular to individual networks that they are applied on, due to the intense scrutiny of data, ensuing pattern recognition, and resulting action of the applied machine learning method. The key ability machine learning is able to provide is that it may optimize congestion control under any circumstance to which it is applied without relying on previously defined methods. The sections below discuss in-depth how these algorithms are developed and implemented in computer networks and will compare them to existing congestion control algorithms that do not utilize machine learning for their optimization of network traffic flow.

5.1 Current Progress

Progress and research towards the development of machine learning based congestion control methods in this section will begin with N. Kato et. al. [8] and the proposed method for traffic control utilizing applied machine learning on heterogeneous networks. This method is then followed by N. SelvaKumar et. al. [7] and their proposed method for applied machine learning using supervised learning methods aided by unsupervised filtering of data. The remainder of this section will in large part focus on the current research surrounding congestion control using machine learning, which itself focuses on reinforcement learning.

5.1.1 Supervised Learning Congestion Control

Supervised learning is well suited for congestion control when paired with predefined thresholds that may be used to differentiate classified data and act using predefined congestion control methods. What supervised learning is able to supplement existing congestion control methods is the ability to actively recognize and discriminate between data points and to classify them as aiding or unaiding in the development of traffic congestion. This information may then be used by aforementioned predefined methods to control the congestion as it is recognized.

N. Kato et. al. [8] propose an algorithm that uses pre-processing via unsupervised learning cluster methods to first group data points and filter outliers, the output dataset is then analyzed for key statistical features which are extracted to create the final processed dataset. This processed dataset is then fed into a supervised learning method that acts to determine the best routing path for all nodes in a system. This method theoretically produces minimal faults and congestion, but is costly in terms of computing power (determining N paths for every node in the network). The method proposed is further ill-suited towards rapid deployment, requiring a pre-existing dataset of previous network traffic. These shortcomings result in a highly effective algorithm when it acts upon the network upon which it is trained for, but also in an algorithm that is unable to adapt to an entirely alien network or for an influx of unusual network traffic. To evaluate performance, the protocol was compared to OSPF in relation to three metrics, Signaling overhead, Average per hop delay, and Total throughput. Signaling overhead was seen to decrease from 55x10^5 down to approximately 13x10^5. Likewise, Average per hop delay decreased from approximately 3200ms to approximately 250ms. Since total throughput takes into account all parameters, an increase was seen from approximately 15.83Mbps to approximately 16.13Mbps.

N. SelvaKumar et. al. [7] propose an algorithm similar to the algorithm noted above in the fact that it utilizes pre-processing, however it contains notable key differences. The proposed algorithm utilizes density-based spatial clustering in its feature extraction and further uses back propagation in its development of the supervised learning classification algorithm. This algorithm is then extended by caching and filtering, which attempts to cache data for similar users and to cache data that is related to current traffic. This caching is intended to reduce future strain on the network such that congestion does not occur. To evaluate performance, the protocol was compared to a historical port-based method. An overall performance score of 97.6% was achieved, a 1.4% increase compared to the port-based method. Although a higher accuracy was achieved, due to runtime analysis not being included, it is unknown how their structure compares to port-based methods in real time execution.

5.1.2 Unsupervised Learning Congestion Control

Unsupervised learning in the context of congestion control simply does not exist in modern or pre-modern implementations. It was seen fit at the time of this paper that a reason be given for the present lack of unsupervised learning methods for congestion control in order to provide a full picture of congestion control methods. The reason unsupervised learning is not seen for congestion control is that unsupervised congestion control seeks to uncover patterns and underlying data structures for the sake of generating information, not generating actionable vectors. While unsupervised learning may be applied in congestion control with an algorithm similar to the one developed by Eric Perraud for threat mitigation, there has yet to be one seen in active research.

5.1.3 Reinforcement Learning Congestion Control

Concerning reinforcement learning are two main branches, reinforcement learning and deep reinforcement learning. Reinforcement learning (RL) in general attempts to assimilate new information and conclusions into a method that is actively developed, where methods such as Aurora [15], DRL-CC [16], SmartCC [17] and NeuroIW [18] attempt to utilize existing datasets to achieve a pre-optimized method for congestion control that is further optimized under new network conditions that it may encounter and be applied to.

Aurora [15] develops a deep neural network (DNN) that consists of state-actions pairs by observing pre-existing network congestion datasets that are supplied during training. Once applied to a network, the algorithm will accept incoming information in the form of states that it has already been trained on, such as latency and sending ratios from neighbouring nodes. These states are then input to the DNN and a resulting action, throttling, inaction, route change, etc. is executed. This method is resilient to large network changes when trained properly on a large dataset, allowing the algorithm to be deployed on networks previously unseen to the algorithm while maintaining low latency and large throughput. Comparison of Aurora is done in relation to TCP Cubic and PCC Vivace [19], with the paper highlighting a decrease in protocol-related latency inflicted upon the network in relation to bandwidth sensitivity and a larger link utilization in relation to packet loss-rates.

DRL-CC [16] builds on previous ideology of utilizing existing datasets, and incorporates several unique mechanisms. With initial reliance on a previously trained actor (algorithm), the actor is further improved upon by a critic-mechanic which is able to critique the actions of the actor under new circumstances by comparing the performance increase or decrease to similar circumstances encountered in the past. In addition to reliance on training with a pre-existing dataset, DRL-CC also incorporates a long short-term memory that allows for further training to be accomplished while simultaneously optimizing congestion for active flowing network traffic. This protocol sees large improvements in goodput (useful throughput data), over existing TCP algorithms, such as LIA and wVegas.

SmartCC [17] develops a Q-network entirely asynchronously from the execution of the protocol. This design was chosen by the developers in order to provide minimal delay during the network execution of the protocol. SmartCC adopts a multipath-congestion approach similar to TQNGPSR seen below. After the algorithm is trained through active observation or dataset learning, the algorithm acts upon a network by choosing and directing traffic to different paths in a heterogeneous network. SmartCC also actively adapts the congestion control window to network congestion. This protocol again sees improvements over existing TCP algorithms, such as LIA and OLIA as mentioned previously in relation to DRL-CC. The improvements noted by [17] are an increase in throughput, a reduction in network jitter, and a reduction in roundtrip time.

NeuroIW [18] focuses less on direct congestion control, instead focusing on the optimization of the initial congestion control window (IW), which is then utilized by existing TCP. This indirect method of optimization is based on rules developed by a previously trained decision-making network, which is adaptive to network conditions through the implementation of a review mechanic that observes the resulting flow completion time in comparison to the previously unadjusted flow completion time. The research suggests that NeuroIW performs better in average flow completion time when compared to a SmartIW [20] algorithm and a static IW of 10.

Moving forwards from DRL to pure reinforcement learning, QTCP [21], TCP-RL [22], and TQNGPSR [23], are all algorithms that focus on the development of a congestion control method when actively involved in network congestion control. These algorithms present a highly dynamic approach that may be applied to various network conditions due to their lack or reliance on existing pre-defined information, as in the case of DRL methods. These algorithms may therefore be implemented at any time to a network without prior information regarding network topology, link state, and other various network conditions and are best suited towards applications where a network is highly dynamic or there does not exist a pre-existing coherent dataset.

QTCP [21] was the first congestion control protocol that actively used a naive reinforcement learning algorithm to determine the optimal congestion control policy. This is done by building off of existing transmission control protocols and passing off control of their rules to a Q-network, which attempts to derive the optimal policies to adopt for individual networks through controlling the congestion control window in regards to the multiple states the algorithm may review as input. The proposed algorithm presents a larger throughput and a lower latency when compared to NewReno.

TCP-RL [22] actively monitors both the congestion control window (CWND) and the congestion control policy that uses said window. This allows both fast adaptation for quick, short traffic flows to ensure optimal network reaction time as well as long term adjustments and preparation for long traffic flows. The CWND is optimized on the fly through reinforcement learning, while the data obtained during ongoing use is integrated into a deep reinforcement learning portion of the protocol that serves to determine the congestion control policy. Research comparisons are drawn from multiple branches of TCP such as an IW of 10 and 200. It is conclusively stated that TCP-RL in the course of the analysis performs better in terms of throughput and a lowered roundtrip time.

TQNGPSR [23] is a development on top of the existing QNGPSR algorithm, which utilizes a Q-network to determine optimal paths in an adhoc network using GPS positioning and various network conditions. TQNGPSR in turn, adds the queue length of the nodes surrounding the active machine so that it may become traffic aware and actively adjust and compare the Q-value (optimality) of routes. When the queue length reaches a threshold that causes the Q-value of a route to fall below an alternative route, this route will then be chosen for new packets that must be forwarded by the active machine. This process continues throughout the lifetime of the network, with the Q-value being actively adjusted as traffic queues are filled in surrounding nodes. In summary, TQNGPSR attempts to control congestion by actively monitoring and comparing surrounding queue lengths, network topologies, and link-reliability. The specific addition of queue lengths in the calculations of the Q-network allow for congestion monitoring and therefore control. TQNGPSR appears as a star above existing flying ad hoc network protocols, improving upon the packet delivery ratio, end-to-end delay, and throughput when compared to existing algorithms such as OLSR and AODV.

In all circumstances observed and surveyed above, it is worth noting that extensive testing was uncompleted for individual algorithms; instead the conclusions developed for the algorithms in question were interpreted from limited test conditions and through comparison to existing previous generation TCP algorithms, such as wVegas, OSPF, traditional TCP, etc., of which all of the above algorithms surpass their individual test cases. In all instances, due to the uneven test circumstances and variance in protocol comparisons, one is unable to make conclusive statements about the benefits of one algorithm over another.

5.2 Future Work

Current implementations exist across a variety of network spectrums from ad hoc (TQNGPSR), to static (Aurora), to dynamically changing (NeuroIW). This vast range of applications clearly implies that congestion control can be actively and with great effect be implemented with regards to, and reliance on, machine learning. In the future due to highly dynamic environments and in order to prepare for situations yet unknown, the next steps towards a unified and optimal congestion control protocol should focus on the implementation of a DRL algorithm that is able to actively optimize itself in a highly dynamic environment. The adaptation towards highly dynamic environments in unison with the ability to rely on previously generated state-action pairs will allow, in the opinion of our team, the best performance in future congestion control governed by machine learning. It is further the opinion of our team that individual algorithms be more extensively tested outside of their proposed test cases, as in the case of all machine learning, an algorithm may too easily become over-trained or optimized for individual use-cases. This testing would further allow for direct comparisons made between machine learning congestion control protocols.

6. Summary

In brief as seen above, supervised is used to an extent that heavily utilizes unsupervised learning clustering methods for the purposes of network management. In addition to the usage of supervised and unsupervised learning applications, focus is also turning towards reinforcement and deep reinforcement learning methods such as

Q-Networks for the purposes of network state-action pairs and conditions. Below we will discuss in brief conclusions that may be drawn from the above field analysis.

Traffic classification at present depends on development of improved supervised and unsupervised learning methods. These methods under current use are used to perform feature extraction such that other algorithms may utilize a dataset that presents differentiating rather than analogous data between different network packets and information. As both clustering methods and feature extraction methods improve, such as advancements in autoencoding or principal component analysis, it is doubtless that these advancements will transfer over towards more efficient and accurate traffic classification for use under machine learning algorithms. While machine learning based traffic classification on its own is not deemed the most useful concept, the field and its research present the unique ability for network data to be identified based on surface analysis rather than existing network packet analysis tools such as WireShark. It is of further importance in noting that traffic classification under machine learning presents the ability to determine the true intent of a packet, such as in the case of a DOS attack that may act under the guise of regular ICMP traffic.

For threat mitigation the field largely focuses on utilization of supervised learning methods such as convolutional networks and belief networks, etc. These networks are often used without dimensionality reduction, however in the case of [11] it is noted that dimensionality reduction over deep auto-encoding is chosen for its analytical benefits. At current standards the methods used provide upwards of 90% accuracy for threat detection upon tested cases. It is of further mention that the implementation of reinforcement learning threat mitigation methods are being explored, such as in the case of ATMoS which developed reinforcement learning agents alongside a threat mitigation framework for network control. While the field possesses greater accuracy for novel threats than existing methods, it is still in its infancy and avenues that may bear greater fruits have yet to be pursued. With the advent of ATMoS introducing scalable reinforcement learning threat mitigation methods, it is doubtless that many more reinforcement learning algorithms will follow.

For congestion control it is clear that the majority of work at current focuses on the development or reinforcement learning and deep reinforcement learning protocols. In addition to the reinforcement learning research, supervised learning based algorithms are also seen achieving greater accuracy than previous generation TCP algorithms. The improvements seen in these algorithms in great part focus on the improvement and modification of TCP and various functions within it, although there are notable instances such as adaptations of GPSR. It is the goal of these algorithms to, instead of developing an entirely new protocol, optimize existing protocols such that they perform as best possible under various circumstances and network conditions. These methods are optimal in that they are able to be deployed on portions of a network such as a business' private network and still allow regular traffic to flow from outside the network without modifying the packets that are involved in the congestion control scheme.

7. Future Work

Future work falls under two categories of issues that are witnessed in this field of research. The first issue is that due to the recency of this field, there is a lack of standard testing procedures including environment variables, comparison metrics, and comparisons themselves. We propose that general rules be developed for the purpose of protocol comparison. These rules could include standardized testing conditions as well as a list of protocols that must be compared to the proposed protocol. Upon completion of these rules, a protocol may be further explored and tested in a wide variety of use-cases to showcase other benefits it seeks to present. The second issue is the broad spectrum of improvements that may be made to existing algorithms which will now be focused on. Below will be three areas that are believed to be of great importance to the improvement of machine learning applications.

Properly training of an algorithm is one of the more difficult steps for a protocol. This can be noticed in both the required time to train and the required data on which an algorithm may be trained. The gathering and human processing of this data requires an immense amount of time in order to have a workable data set, and the time training the algorithm is time that is not being spent deployed on the network. This issue is overcome in the instance of both naive reinforcement learning algorithms and unsupervised cluster-recognition algorithms, however this naiveness can develop poor initial responses to congestion and threats in a network. What may be best focused on to mitigate this issue is the development of highly adaptable pre-trained protocols such as those based on supervised or deep reinforcement learning. This development will likely come with advances in the field of machine learning itself but it goes without saying that further human development specific to the integration of machine learning on computer networks may bear positive results specific to algorithm adaptivity to extreme network conditions.

Execution time of a protocol is of further concern for network administration. While congestion control has the failsafe of simply allowing congestion to occur, threat mitigation must be fully aware and able to respond to threats before they are allowed to spread throughout a network. This hypervigilance a threat mitigation protocol must possess has the positive effect of actively filtering out all threats, while also having the negative effect on traffic flow time due to its screening requirements. As seen by the requirement that Perraud [10] develop his own algorithm in order to have one that is efficient enough that it may be run on nearly any CPU without incurring large calculation times. It is imperative that machine learning implementations be done in such a way that they have minimum effect upon their target network. In the future, research may be focused on maintenance of low computation, high efficiency protocols that are able to work effectively on their given task without negatively impacting traffic RTT.

Finally, as always for cutting edge research and in fact all research that requires precision, accuracy is an area that stands to be constantly improved upon. At current, accuracy for supervised and deep reinforcement learning algorithms depends on their training, wherein they may perform poorly if they are introduced to environments for which they have little experience observing and acting upon. An improvement that may be possible in future works are enhanced datasets paired with algorithms that are able to actively differentiate between various network conditions accurately and precisely. Evidence of room for improvement can be seen in the algorithm proposed in [13] possessing a 97.5% detection accuracy or in Perraud's algorithm [10] which posses 100% accuracy for detecting threats while also being over zealous in its classification, causing less than 100% of benign traffic to be correctly classified. While the algorithms presented today are reasonably accurate under their current test and use cases, there is still room for improvement that may be seen in future work and research.

8. Conclusion

In conclusion, it is clear that steps to implement machine learning into network communication control and management have clear benefits. These benefits are widely seen in machine learning allowing for greater accuracy when detecting, classifying, and preventing threats from infiltrating a network and in the numerous congestion control algorithms that may be utilized for optimization of existing network connections. While network classification may not appear to be a prevalent topic at the time of writing, it is of comprehensible importance in the deployment of almost all the algorithms discussed due to their needs for classified data in order to act upon the networks under management. While congestion control and threat mitigation may see individual advancements, it is clear that steps towards improving traffic classification methods will doubtlessly see improvements following in both aforementioned fields. All things considered, the field for network communication and management utilizing machine learning is alive and well, presenting ever new and improving solutions that will unquestionably benefit networks of the future as well as the present.

Appendix

Authors

Jacob Thom is a fourth year Computer Science student at the University of Victoria and is pursuing a focus in Computer Communications & Networks. His main areas of interest include Machine Learning, Datamining, Network Security, and Finite Mathematics.

Dafydd Foster is a fourth year Computer Science student at the University of Victoria. His interests include Machine Learning, Distributed Systems, Datamining, and Embedded Systems.

References

[1] C. Knowles, “Internet outages drastically increased during COVID-19 lockdowns, report finds,” SecurityBrief Asia, 07-Aug-2020. [Online]. Available: https://securitybrief.asia/story/internet-outages-drastically-increased-during-covid-19-lockdowns-report-finds.

[2] A. Aldweesh, A. Derhab, A. Z. Emam, “Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues,”Knowledge-based Systems, vol. 189, art. 105124, Feb. 2020. [Online]. Available: ScienceDirect, https://www.sciencedirect.com.

[3] S. Gamage, J. Samarabandu, “Deep learning methods in network intrusion detection: A survey and an objective comparison,” Journal of Network and Computer Applications, vol. 169, art. 102767, Nov. 2020. [Online]. Available: ScienceDirect, https://www.sciencedirect.com.

[4] T. Zhang and S. Mao, "Machine Learning for End-to-End Congestion Control," IEEE Communications Magazine, vol. 58, (6), pp. 52-57, 2020. . DOI: 10.1109/MCOM.001.1900509. [Online]. Available: IEEE Xplore, https://www.ieee.org

[5] Y. Zhao, Y. Li, X. Zhang, G. Geng, W. Zhang, Y. Sun, “A survey of networking applications applying the software defined networking concept based on machine learning,” IEEE Access, vol. 7, pp. 95397-95417, July 2019. [Online]. Available: IEEE Xplore, https://www.ieee.org.

[6] M. Furdek et al, "Optical network security management: requirements, architecture, and efficient machine learning models for detection of evolving threats [Invited]," Journal of Optical Communications and Networking, vol. 13, (2), pp. A144-A155, 2021. [Online]. Available: IEEE Xplore, https://www.ieee.org. [Accessed: Feb 10, 2021].

[7] N. SelvaKumar, M. Rohini, C. Narmada, M. Yogeshprabhu, "Network Traffic Control Using AI," International Journal of Scientific Research in Network Security and Communication, Vol.8, Issue.2, pp.13-21, 2020. [Online] Available: IJSRNC, https://www.ijsrnsc.org.

[8] N. Kato et al, "The Deep Learning Vision for Heterogeneous Network Traffic Control: Proposal, Challenges, and Future Perspective," IEEE Wireless Communications, vol. 24, (3), pp. 146-153, 2017. [Online]. Available: IEEE Xplore, https://www.ieee.org.

[9]“Security Event Manager,” SolarWinds. [Online]. Available: https://www.solarwinds.com/ security-event-manager. [Accessed: 12-Apr-2021].

[10] Perraud, Eric. (2019). Machine Learning Algorithm of Detection of DOS Attacks on an Automotive Telematic Unit. International journal of Computer Networks & Communications. 11. 27-43. 10.5121/ijcnc.2019.11102. [Online] Available: www.researchgate.net.

[11] F. Farahnakian and J. Heikkonen, "A deep auto-encoder based approach for intrusion detection system," 2018 20th International Conference on Advanced Communication Technology (ICACT), Chuncheon, Korea (South), 2018, pp. 1-1, doi: 10.23919/ICACT.2018.8323687.

[12] D. Kwon, K. Natarajan, S. C. Suh, H. Kim and J. Kim, "An Empirical Study on Network Anomaly Detection Using Convolutional Neural Networks," 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, 2018, pp. 1595-1598, doi: 10.1109/ICDCS.2018.00178.

[13] M. Z. Alom, V. Bontupalli and T. M. Taha, "Intrusion detection using deep belief networks," in 2015, . DOI: 10.1109/NAECON.2015.7443094.

[14] I. Akbari, E. Tahoun, M. A. Salahuddin, N. Limam and R. Boutaba, "ATMoS: Autonomous Threat Mitigation in SDN using Reinforcement Learning," NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 2020, pp. 1-9, doi: 10.1109/NOMS47738.2020.9110426.

[15] N. Jay, N. H. Rotman, P. Brighten Godfrey, M. Schapira, and A. Tamar, “A Deep Reinforcement Learning Perspective on Internet Congestion Control,” Proceedings of the 36th International Conference on Machine Learning, vol. 97, Jun. 2019.

[16] Z. Xu et al, "Experience-Driven Congestion Control: When Multi-Path TCP Meets Deep Reinforcement Learning," IEEE Journal on Selected Areas in Communications, vol. 37, (6), pp. 1325-1336, 2019.

[17] W. Li et al, "SmartCC: A Reinforcement Learning Approach for Multipath TCP Congestion Control in Heterogeneous Networks," IEEE Journal on Selected Areas in Communications, vol. 37, (11), pp. 2621-2633, 2019.

[18] R. Xie, X. Jia and K. Wu, "Adaptive Online Decision Method for Initial Congestion Window in 5G Mobile Edge Computing Using Deep Reinforcement Learning," IEEE Journal on Selected Areas in Communications, vol. 38, (2), pp. 389-403, 2020.

[19] Dong, M., Meng, T., Zarchy, D., Arslan, E., Gilad, Y., Godfrey, B., & Schapira, M. "PCC vivace: Online-learning congestion control," Symposium on Networked Systems Design and Implementation, vol. 15, pp. 343-356, 2018. [Online] Available: https://pbg.cs.illinois.edu

[20] X. Nie *et al*, "Reducing web latency through dynamically setting TCP initial window with reinforcement learning," in 2018, . DOI: 10.1109/IWQoS.2018.8624175.

[21] W. Li et al, "QTCP: Adaptive Congestion Control with Reinforcement Learning," IEEE Transactions on Network Science and Engineering, vol. 6, (3), pp. 445-458, 2019.

[22] [1] X. Nie et al, "Dynamic TCP Initial Windows and Congestion Control Schemes Through Reinforcement Learning," IEEE Journal on Selected Areas in Communications, vol. 37, (6), pp. 1231-1247, 2019.

[23] Y. Chen et al, "A traffic-aware Q-network enhanced routing protocol based on GPSR for unmanned aerial vehicle ad-hoc networks," Frontiers of Information Technology & Electronic Engineering, vol. 21, (9), pp. 1308-1320, 2020. [Online] Available: Springer, https://www.springer.com/us