Implementing a Fused Machine Learning Model for the Provision of Smart Health Care in MANETS.

Kirori Mindo[1], Moses M.Thiga[2], Simon Maina Karume[3]

kirori@kabarak.ac.ke, mthiga@kabarak.ac.ke, smkarume@gmail.com

KabarakUniversity, P.OBox Private Bag 20157 Kabarak Nakuru, Kenya

ABSTRACT

Mobile Ad-Hoc Networks – MANETs are prevalent in healthcare monitoring of high blood pressure, high cholesterol levels and various heart conditions and cardiac misnomers like syncope, third murmurs and atrial fibrillation. These irregularities that cause mysterious fainting, unexplained stroke, heart palpitations and atrial fibrillation need to be monitored remotely, accurately and effortlessly. However, the growth and provision of the internet of things based smart healthcare monitoring has faced various security obstacles, primarily security. The characteristic mobility of these health monitoring devices as well as their inherently dynamic network topology, causes the connectivity structure to change frequently and unpredictably. Further, these smart devices have limited resources in storage, processor capability and memory; thus, these weaknesses and inherent nature makes them subject to attacks like Denial of Service (DoS) attacks. There is need to provide resilient security methodologies that do not require enormous computing resources. While entry prevention is the most viable disposition, it is not always possible to stop unauthorized access. Thus, it is critical to investigate the use of machine learning-based intrusion detection to buttress and provide sufficient security against DOS and other attacks in MANETs. Various anomaly-based intrusion detection systems employ varying techniques to identify anomalies in the context of diverse and valid variables. Most of these techniques, however, fail to capture and take account the physiognomies of MANETs. In the intervening time, usage of the internet of things in the provision of smart healthcare is expanding and the inherent risks snowballing. Attacks aimed at MANETs are increasing to an alarming extent. This study employed a fusion of machine learning techniques through both simulation and a running prototype to achieve a more resilient intrusion detection system. The study was implemented and evaluated on a MANET environment on both Linux NS 2 and further implemented on a network of Smart wearable devices and Raspberry Pi. This study contributed to the body of knowledge in the field intrusion detection systems through ubiquitous learning.

Keywords: MANET, Smart Healthcare, Intrusion Detection Systems, Machine Learning, Fused, Internet of Things.

INTRODUCTION

The use of smart devices in the provision of healthcare provides numerous benefits. Use of technology in the healthcare profession has generally led to faster diagnosis, lower costs, health workers and researcher(s) collaboration, reliable services, efficient and effective healthcare systems as well (Reddy et al., 2018). The provision of smart healthcare services is dependent on the Internet of Things that run on MANETs. While it is particularly indispensable, the security of the systems and data remains a critical challenge that hinders the accelerated adoption of smart health care (Iyengar, Kundu, & Pallis, 2018).

What are MANETs?

Mobile Ad Hoc Networks (MANET) is a network of physical electronic devices that are embedded with electric and electronic components, software, sensors, actuators, radios and rooted within everyday tools or machinery like home appliances, vehicles, wearable devices like watches, doors, traffic lights, switches and other items (Alagoz et al., 2017). These devices have radios that enable connectivity to networks which permit these devices to connect and exchange data with other devices or networks (Vermesan et al., 2011). The Internet of Things system allows interrelated computing devices, mechanical and digital machines, objects, animals or people that have unique addresses that gives them the ability to transfer data over a network effortlessly without human-to-human or human-to-computer intervention (Agrawal, & Vieira, 2013).

While each of the internet of things is uniquely identified using layer 2 addresses, within its embedded computing, firmware or operating system, the device also has the ability to inter-operate within the Internet infrastructure as well as local network (Savolainen, Soininen, & Silverajan, 2013). The MANET enables devices to be sensed, connected, communicated or controlled remotely over the existing wireless network infrastructure (Hsiao, Lian, & Sung, 2016). This provides an enabling opportunity for more direct integration with the physical world and computer-based systems. These systems result in improved ease, efficiency, accuracy and economic advantage as well as reduced human involvement in their control and usage (Nigam, Asthana, & Gupta, 2016). Once MANET is amplified with sensors and actuators, this wireless technology becomes an exquisite and ubiquitous manifestation of the common cyber-physical computerized systems commonly found in working, business, industrial or agricultural fields. These MANET are also encompassed and adopted by other technologies such as smart homes, smart grids, smart cities, virtual power plants, smart agriculture, smart weather and intelligent transportation (Nigam, Asthana, & Gupta, 2016). These "Things", refer to a wide assortment of electronic devices such as wearable technologies, health and heart monitoring implants, biochip transponders on wildlife and domesticated animals, CCTV cameras that stream audiovisual data, car tracks, DNA analyzing devices in the environment, food, disease, pathogens, buildings, and even field operation devices used in firefighting, search and rescue operations (Bedi, Venayagamoorthy, & Singh, 2016).

These things are a tangled mix of hardware, software, data and service. Consequently, the consequences of embedding the internet of things with minuscule addressable devices or machine-readable sensors would be the transformation of business, security and daily activities. The capability to interact remotely with devices based on a person's immediate needs greatly eases and improves quality of life (Stankovic, 2014). The interconnection of these devices allow for generation of data from remote devices to other objects includes the notion of the connecting the physical world with a virtual world into a multi-level oriented architecture with the nature and devices at the bottom level, the Internet, sensor network, and mobile network, and intelligent human-machine communities at the top level. This architecture disperses users and enables them to accomplish tasks ubiquitously as well as solve everyday problems by using the ad-hoc neural

.

network. This network enables the active flow of data, information, knowledge, material, energy and services in the global troposphere (Wu, Meng, & Gray, 2017). This gravitating superlative model envisioned the development and growth of the Internet of Things (Al-Fuqaha et al., 2015).

MANETs are by their very nature mobile and dynamic. Thus inherently, they lack neither a fundamentally uniform coordination nor rigid hierarchical topology architecture. This, coupled with the lack of a centrally coordinated security system, makes these devices especially vulnerable to attacks as opposed to wired networks. Smart devices have limited resources in low storage, low memory and limited processing power. Their inherent nature incapacitates their ability to shield themselves against eavesdropping, malicious attacks, packet-sniffing and other security threats. The confidentiality, or availability of systems and data becomes compromised as a result.

Denial of Service attacks is a common threat in MANETs that denies users from accessing the system(s) and information when and if they require it. This DOS/DDOS attack targets a device by using malicious unwanted response requests, thereby draining resources and rendering the device unable to respond to genuine user requests (Hui, Kim, & Wang, 2017). Two strategies can be employed to identify such DoS attacks; Signature detection and Anomaly detection. The Signature analysis techniques employ the tactic of consolidating information in a manner similar to expert systems methodology. The signature approach, however, uses the information that is consolidated in a different way, by deciphering and breaking down data into a series of appraised events thereby decreasing the alarm threshold of the intrusion system (Kumar, Mangathayaru, & Narsimha, 2016). Efficiency is the utmost trait of this signature-based analysis technique which has enabled its implantation in the market as a viable enterprise security system. The solution, however, has a major bottleneck in the requirement for regular updates so as to protect the network from newly discovered threats (Iqbal et al., 2016).

Any MANET implementation, therefore, needs to embed security and privacy by design which should be part of any MANET project, use case or deployment. Leveraging on MANET and bio-medical data aims to improve and reduce errors and costs. Making sure data and devices do not get exposed or used for the wrong reasons is a key proposition for any Smart Health implementation. The personal or confidential nature of health data makes it considerably challenging when implementing MANET as the threshold for security and privacy is much higher, even supported by regulation (Munns, & Basu, 2015).

Security of smart healthcare devices that provide mission-critical support in healthcare is extremely vital. Devices in smart health include but are not limited to wearable heart monitors, body sensors and pacemakers. These devices are primarily adorned so that deteriorating patient's medical condition is observed and identified or alerted in time. However, these devices, which run on MANETs, are such that their physiognomy lacks the adequate capability to devise robust systems to shield themselves against eavesdropping, malicious attacks, packet-sniffing and other security threats, especially DDOS. DDOS attacks can conceal deteriorating health risks from discovery by both the patient and health specialist thus can lead to death, immobility, permanent or impaired disability. A patient’s worsening condition might not be alerted to both the patient and health specialist as envisaged. Medical concerns within hospitals include unexplained

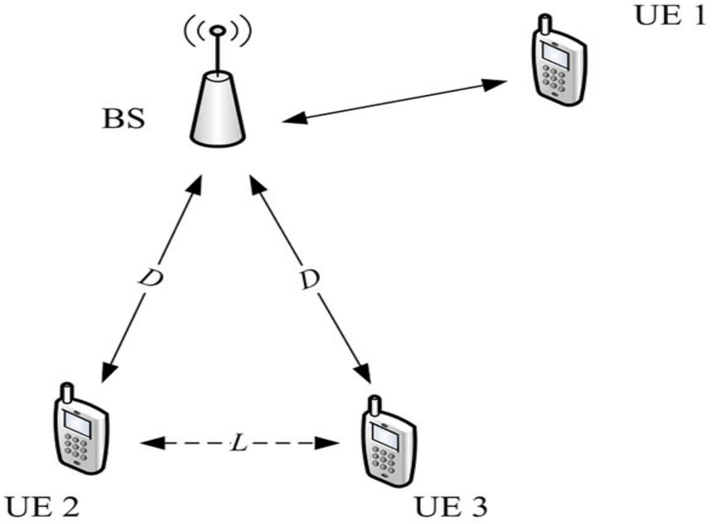
.

fainting, unexplained stroke, heart palpitations and atrial fibrillation that need to be monitored remotely, accurately and effortlessly. Cases of DDOS attacks against smart health devices are on a rapid rise. Most IDS techniques fail to capture and take account of the characteristics of MANETs, which malicious attacks exploit. Existing intrusion detection methods are weak in identifying anomalous activity within a mobile ad-hoc wireless network.

LITERATURE REVIEW

History of the Internet of Things

The terminology "the Internet of things" was coined in 1999 by Kevin Ashton, a technology pioneer and assistant brand manager at Procter & Gamble (Mehta, Kale, & Utage, 2017). The AutoID Center consortium at MIT formally began research on the Internet of Things (Michael, 2017) The Internet of things industry has then evolved into a convergence of numerous communication and wireless technologies, that enable ubiquitous wireless communication, embedded sensors, real-time analytics, artificial intelligence and machine learning, and embedded systems (Norman, 2017). The genesis enabler for the MANET concept was that of a network of smart devices which was in 1982, where a modified Coke dispensing machine at Carnegie Mellon University was extraordinarily the first device to have an Internet connection (Saha, Mandal, & Sinha, 2017). This appliance provided information on the available stock count and whether sodas were chilled (Breur, 2015). Mark Weiser's vision in 1991 presented through a seminar paper titled "The Computer of the 21st Century" drew ideas on the ability to have ubiquitous computing (Kušen, & Strembeck, 2017). The dream was further driven through various academic venues such as UbiComp and PerCom, which enabled a contemporary vision of the Internet of Things (Kaur, & Saini, 2017). Later in late 1994, Reza Raji described the concept of moving small packets of data between a large set of nodes enabling home automation, through home appliances as well as automation of entire factories. Further, several companies between 1993 and 1996 anticipated solutions in the MANET spectrum. Nonetheless, it's only in 1999 that the MANET industry started gathering traction. In 1999 Bill Joy presented to the World Economic Forum his revolutionary Device to Device (D2D) communication technology that allowed the exchange of data between two devices, at Davos (Borgohain, Kumar, & Sanyal, 2015). Figure 2 below shows the underlying communications systems for Device-to-Device Communication.



.

Figure 1: Device-to-Device Communication Underlying Cellular Communications Systems (Janis et al., 2009).

The concept of the Internet of things was popularized in the early1999, at the MIT's Auto-ID Center and further through related market-analysis reports and publications (Sundmaeker, Guillemin, Friess, & Woelfflé, 2010). Use of Radio-frequency identification (RFID) made a great contribution where one of the founders of the Auto-ID Center - Kevin Ashton espoused and broke ground for the Internet of things by providing and leading in MANET research. Kevin Ashton phrased MANET as a scenario where all objects, machines and people in daily life were embedded with identifiers, thus allowing computers to manage and audit them. In addition to the use of RFID in tagging most of the internet of things, this phenomenon could be accomplished through technologies such as barcodes, near field communication, Quick Response codes and digital watermarking (Vongpradhip, & Rungraungsilp, 2012).

Gartner projected that 6.4 billion Internet of Things would be in use in 2017. This forecasts that the number will grow tremendously, three-fold to 21 billion by the year 2020. Yu et al. (2015) estimated that the number of MANET devices deployed will grow exponentially from 5 Billion in 2015 to 25 Billion by the year 2020. Other studies in 2013 estimated that 9 billion MANET devices were in use and forecasted the numbers to grow beyond 24 billion in 2020 (Gubbi, Buyya, Marusic, & Palaniswami, 2013). Other experts have put evaluations on the MANET indicators with the ecosystem consisting of 30 billion objects by 2020, and further a 17 to 32 per cent annual growth, thus the MANET industry will grow to be more than a trillion-dollar market before 2020. While various radio wavelength technologies are in use, the most common are Light-Fidelity, Near-field communication (NFC), QR codes and barcodes, Radio-frequency identification (RFID), Thread, Wi-Fi, Z-Wave and ZigBee. By 2011, there were 2 billion Wi-Fi certified devices in use for various connectivity purposes (Bartoli et al., 2011). Bluetooth has the lions share with 3 billion Bluetooth enabled devices in the market in 2014; in addition, over 10 billion Bluetooth devices will be available on the global market by 2018. Wireless Sensor Networks mainly exhibit a small packet data size that propagates a packet size of 127 Bytes and 81 octets for data packets. WSN support both 16-bit short as well as the IEEE 64-bit extended MAC addressing scheme. WSN also have low throughput and low consumption of bandwidth as well, with data rates of 250 kbps, 40 kbps, and 20 kbps for the physical layers. These technologies support logical topologies like star topology; however, the most common logical topology is the mesh. WSN also support high device density, with up to 60,000 wireless devices connecting in a network. In addition, the devices use low power by muting its sending/receiving capability when not required. Devices in such networks are classified either into a full-function device (FFD) or reduced function devices (RFD).

Common Attacks in the Internet of Things

The following some of the common attacks on the internet of things as presented by various authors

i. Leakage of Information (Confidentiality)

Data and information collected and transmitted by the smart devices within a MANET wireless sensor network is susceptible to leakage. Data and information from these devices is easily

.

leaked since there lacks sufficient data encryption that is applied either between gateway and sensors or between the sensors themselves. In addition, user authentication to prevent unauthorized access and/or enable detection of unwanted and unauthorized parties is often weakly implemented (Rath et al., 2018).

ii. Denial of Service and/or Distributed Denial of Service (Availability)

This is a common attack that denies users from accessing the system(s) and information when and if they require it. This DOS/DDOS attack targets a device by using malicious unwanted response requests, thereby draining resources and rendering the device unable to respond to genuine user requests. While no data is leaked or exposed, it is very disastrous as it makes systems unusable and renders data/information un-useful as a result for the period of the attack (Dhindsa, & Bhushan, 2019).

iii. Falsification (Integrity)

This attack happens when a wireless device is in communication with the gateway and the attacker successfully captures the collect packets in transition and alters the fields containing routing information. As a result, the attacker can access the information therein and alter, leak or destroy the data/information as a whole. Most SSL mechanisms have the capability to protect against this type of attack, while unauthorized devices that gain access should be entirely blocked. Most of these attacks happen as passive eavesdropping and/or traffic analysis. Hostiles silently listen to communication (Ngomane, Velempini, & Dlamini 2018)

Intrusion Detection Models for the MANET

Intrusion Detection System is a security measure that can be installed on a network to prevent attacks from happening. The IDS allows network administrators to detect individuals trying to compromise the system so that they retrieve information from it. There are various activities that the administrators can detect in order to identify it. This includes security policies violation (Chaudhary, & Shrimal, 2019). The IDS works best because it designed in a manner that enables it to detect the vulnerabilities on the system in which it is installed. For example, it can work on the basis of previous attacks that affected the network and work backwards to eliminate the chances of another similar attack. The IDS can detect attacks using various methods. For example, it can be done through signature-based detection. In these patterns are studied and compared to previous events or attacks and then identifies new threats. As a result, the system administrators can be able to identify new threats and other kinds of threats that the network is vulnerable to. An IDS is made of three basic components that include Network Intrusion Detection System (NIDS), Network Node Intrusion Detection System (NNIDS) and Host Intrusion Detection System. Each of these components plays a very vital role in securing networks (Benkhelifa, Welsh, & Hamouda, 2018).

The Network Intrusion Detection System works by first analyzing the traffic on the network. It then identifies possible threats with those attacks that are already registered on its library. The Host Intrusion Detection System, on the other hand, captures the image of the entire system file set and then compares it with the previous picture. If there is a difference at all, then it alerts the system's administrators who then comes and stops the possible attack. There is also a Cloud Intrusion Detection system that is used for public environments. There are two general types of

.

Intrusion Detection Systems. They are host-based intrusion system and the network-based intrusion system. Each of those two systems has sensors that are aligned to the type of intrusion system. There are sensors on a network-based IDS that monitor streams of traffic (Kenkre, Pai, & Colaco, 2015). The network-based IDS have various advantages and disadvantages. The following are the examples of the advantages that are aligned to this kind of a network; there is a lower cost of ownership when organizations and governments have the network-based IDS. This is because the traffic on the network is monitored as a whole (Gai, Qiu, Tao, & Zhu, 2016). This means that the tendency of loading software on each host on the network is omitted. It is easier to deploy a network-based network. One of the advantages aligned to this is that the installation of that network does not affect the existing infrastructure. Detecting a network-based attack on this kind of network is easier. This is because the network-based IDS have sensors that check all the packets and identify any threat that may exist on the network. Using a network-based IDS is also advantageous because it has real-time detection and quick responses to any kind of attack that might face the network. The host-based intrusion detection system has various advantages that are tied to it (Liu et al., 2018). The kind of system can be able to give feedback to the success or failure of the attacks on that network. This is because the Host-based intrusion detection system contains logs of all activities that have taken place. This kind of IDS can also be able to monitor the activities that affect a given network were it is installed. Another advantage is that the host-based IDS is that it can be able to detect the attacks that have been caused on the Network-based IDS and gone unnoticed. Network-based IDS sensors cannot, for example, detect when an unauthorized user makes changes on a network. The sensors based on the host are more superior to the other kind of sensors. These kinds of IDS also have a near real-time detection and response to attacks and threats to attacks. This is very important for administrators because they can be able to handle attacks way before they are actually launched (Jose et al., 2018).

The host-based Intrusion detection sensors are installed inside the host servers or machines that play host to them. This means that there is no additional hardware that is required in order to install this kind of IDS. When a comparison is made between the host and network-based sensors, the host-based sensors are way cheaper. This means that the cost of entry to this kind of IDS is cheaper. From the sound of the advantages of these two kinds of IDS, we are prompt to assume that every institution out there needs either of these IDS. However, each of these IDS comes with disadvantages that might limit its performance, and it is important to know each of them (Marteau, 2019). The IDS technology is advancing on a daily basis and therefore organizations that acquire either of them should ensure that their system is up to date so that it can be able to handle even the most recent kinds of threats that can be launched on a given network. Having an IDS system on a network is not the solution to preventing all kinds of attacks. The success of these systems depends heavily on the way the IDS sensors are deployed on a given network. Therefore system administrators should ensure that the deployment procedure is done and achieved in the manner that they are supposed to take place. The IDS technology also is a reactive activity, not a proactive. This simply means that the IDS technology heavily relies on previous attack patterns. The technology cannot work independently. However, IDS technology is very important for any organization that seeks to secure itself right from the network level (Taher, Jisan, & Rahman, 2019). A lot of information can be secured through the process, and this is what each organization seeks to achieve at the end of the day. It is important to put in place better identification and strong authentication processes.

.

Implementation of MANET Anomaly-based Intrusion Detection Model for Health Care

An anomaly-based intrusion detection system, monitors and alerts intrusions and misuse by observing activity that falls out of normal system operation. This is in contrast to signature-based systems, which can only detect attacks for which a signature has previously been created. Anomaly-based intrusion detection has the capability to identify unknown intrusions as well as zero-day assaults. The strength of this emerges from the capability of ABIDS to model standard operation disposition of a network and further identify deviations from the baseline. ABIDS can also be specifically configured to suit a particular network thus making it challenging for a previously successful attack in one network to be replicated in a unique setting (Gai, Qiu, Tao, & Zhu, 2016). Anomaly Based intrusion detection systems can implement different methodologies in either; an artificial intelligent knowledge-based detection, statistical anomaly detection, data-mining based detection or a machine learning-based detection algorithm.

a. Statistical ABIDS Model (SABIDS)

A Statistical intrusion detection model is a common technique for identifying attacks and intrusions in a network. Statistical based anomaly detection techniques employ statistical values and statistical assessments to conclude whether the observed performance departs considerably from the expected norm (Zaidi et al., 2016). These statistical anomaly detection systems rely on the basis of a quasi-stationary activity, which is rare for most of the data processed by anomaly detection methods. Further, the SABIDS learning process is long and takes before attaining accuracy and effectiveness. In addition, SABIDS have a conundrum in setting the right alarm volume. A volume too high might not identify attacks, while a low alarm volume might result in numerous false notifications (Moustafa, Creech, & Slay, 2017).

b. Operational Model or Threshold Metric Model

Operational Model also referred to as the Threshold Metric model, is founded on the hypothesis that abnormal activity can be recognized by comparing a stream of activity against a predefined limit. On the basis of observed activity over a phase of time, an alarm can be raised. The methodology is applied; particular statistics are commonly related to network intrusions. An Adaptive Threshold Algorithm can be combined in this case scenario as a child model. The Adaptive threshold algorithm is a simple and straightforward methodology which evaluates whether the amount of data over a given interval meets a set threshold (Saikumar et al., 2017).

c. Markovian Process Model or Marker Model

The Markovian/Marker methodology is co-joined together with data values so as to conclude on the regularity of a specific occurrence, on the basis of prior events. This model symbolizes every captured data as an isolated case and exploits a state transition method to establish whether the observed occurrence is normal on the basis of prior events. This model is principally advantageous if the sequence of occurrences is predominantly significant. This model is based on two major procedures the Markov chains and the Markov models (Almusallam, Tari, & Zomaya, 2017).

.

d. Statistical Moments or Mean and Standard Deviation Model

The Statistical Moments or Mean and Standard Deviation Model implements statistical prediction and evaluation from the norm on the basis of current values measured against a spread of possible scenarios. The statistical moments sub-model evaluates and concludes that a particular occurrence which goes beyond a set alarm value is anomalous. This method comprehends device instances without the prior knowledge of the devices traffic behaviours. This methodology is very flexible and has the ability to determine anomalous activity without prior briefing or configurations. It is, however, a very complex model to implement and build (Kumar, & Venugopalan, 2017).

e. Multivariate Model

This Multivariate Model used to monitor and detect intrusions on the basis of two or more behavioural occurrences. The Multivariate Model thrives in occurrences of two or more behavioural occurrences that allow identification of possible irregularities in cases of complex conditions with multiple constraints. This Multivariate Model, when augmented with statistical methods like chi-square, produces improved results with low false alarm experienced as well as a high detection rate. In this methodology, anomalous activity is identified fast. This model, however, is computer-intensive with large volumes of statistical processes required so as to evaluate events accurately (Kolhe et al., 2016).

g. Time Series Model

The TimeSeries Model identifies intrusions through a process of evaluating the sequence and time taken to perform various tasks in a networked system. An occurrence is marked as normal if the metrics are lower than the threshold, while an occurrence is marked as anomalous if the metrics are observed to be higher than the threshold. The Time Series methodology is flexible since it adapts and modifies itself on the basis of user actions. The alarm is set off by activities that exhibit substantial departure from the regular disposition (Wang et al., 2016).

h. Data Mining Based Approach

The Data Mining Based Approach is useful when used to identify external malicious traffic coming into the network. It, however, is a weak mechanism for identifying internal cases of attacks. This data mining method is crucial in excluding regular occurrences from raising the alarm, thus enabling security admins to only dedicate time to managing actual network attacks. The data mining approach has the capability of identifying false alarms as well as irregular signatures, thus enabling only the actual anomalous activity being identified and acted upon. This Data Mining based method uses two major techniques; clustering of traffic into groupings and identifying normal occurrences while facilitating the discovery of attacks. (Sahasrabuddhe et al., 2017).

i. Association Rule Discovery

The Association Rule discovery is a common methodology while albeit slow, uses the correlation between various elements to identify anomalous activity. This method is used in a "market basket analysis" case scenario that identifies irregularities in the purchasing conduct of supermarket clients. Also referred to as Boolean association rules, this technique tries to identify various arrays of items within the market that consumers regularly buy consequently in every

.

purchase. Disadvantages of this methodology are that it exponentially proliferates the occurrences as the number of elements grows (Kong, Jong, & Ryang, 2016).

j. Knowledge-Based Detection Technique

Knowledge-based detection methodology is a flexible technique which is applied in both anomaly-based intrusion detection systems as well as signature-based systems. This technique captures and stores known intrusions and network threats. This stored information is then employed to mitigate future intrusions as well as raise the threshold alarm. Occurrences that do not trigger the threshold alarm are treated as safe events (Kevric, Jukic, & Subasi, 2017).

k. State Transition Analysis

The State transition analysis methodology is an open-source technique that reviews possible intrusions through objectives and transitions. These state transition diagrams provide pictorial illustrations of events that an attacker can successfully perform so as to attack a network. The sequence of occurrences undertaken during an attack towards a network is recorded for every compromised state. State transition illustrations recognize the necessities for every intrusion as well as causes for the attack success. These illustrations enable the identification of key occurrences that enable an intrusion possible (Le Dang, Le, & Le, 2016).

l. Expert System

An experts system is a form of artificial intelligence, through which a computer system uses to imitate the decision-making capability of a human. This technique integrates a knowledge-based intrusion-detection methodology. The expert system comprises a set of rules which describe attacks. Audit events are thereafter translated into facts carrying their semantic signification in the expert system, and the inference engine draws conclusions based on these rules and facts. This method increases the level of abstraction of the audit data by attaching semantic to it. Expert systems are integrated into both signature-based and anomaly-based intrusion detection systems as well (Folorunso, Ayo, & Babalola, 2016).

m. Signature Analysis

The Signature analysis techniques employ the tactic of consolidating information in a manner similar to expert systems methodology. The signature approach, however, uses the information that is consolidated in a different way, by deciphering and breaking down data into a series of appraised events, thereby decreasing the alarm threshold of the intrusion system. Efficiency is the utmost trait of this signature-based analysis technique which has enabled its implantation in the market as a viable enterprise security system. The solution, however, has a major bottleneck in the requirement for regular updates so as to protect the network from newly discovered threats (Iqbal, & Calix, 2016).

n. Machine Learning-Based Detection Technique

Machine learning is the capability of an application or network system to acquire and advance its security capability by consolidating data and information and building a concrete algorithm as a result. Machine learning-based detection is reliant on the construction of a system which expands and progresses the ability to protect the network on the basis of improving on prior performance. This technique enables the machine-based system usable in wide and varying case scenarios.

.

However, this machine learning intrusion detection system gobbles up system resources thus can partake a huge amount of memory, bandwidth and CPU time (Buczak, & Guven, 2016). Machine learning intrusion detection systems use either in artificial intelligent fuzzy logic, artificial neural networks, Bayesian techniques, genetic algorithms, support vector machines and Bayesian systems.

The Bayesian Approach technique uses illustrational diagrams to reveal possible interactions between various devices and resources in a network. This system can be used in cases where data flow is unpredictable in nature and intrusion case scenarios cannot be pre-determined (Kabir, Onik, & Samad, 2017). While the Bayesian Approach is a relatively new technique in the field of intrusion detection, its grouping and combination with statistical methodologies have risen to efficient network security solutions. The only challenge with this technique is that it is computer-intensive in nature and can only work best when integrated with a Pseudo-Bayes technique so as to enhance the anomalous intrusion detection system's capability to identify new types of intrusions with a low threshold. The Bayesian Approach has the ability to spontaneously accommodate traffic streams with missing entries that would otherwise make it difficult to decipher an attack. This technique is superlative for the consolidation of an existing intelligent formation and actual traffic flow (Mir, Khan, Butt, & Zaman, 2016).

o. Neural Networks Technique

A neural network is an interconnection of various devices within a network, where the computational output of one device forms the input of another. Intrusion detection systems implementing the neural network approach have the capability to anticipate consequent user actions by any device or user in the network. This technique has the capability of conceptualizing an occurrence that results in anomaly detection. Neural networks in the construction of intrusion detection systems reveal an efficient substitute for statistical methodologies. Neural networks are commonly found in anomaly-based intrusion detection systems (Roy et al., 2017).

p. Fuzzy Logic Approach

Fuzzy logic methodologies have the capability to handle significantly huge amounts of traffic and domain constraints, especially in a scenario with cases of data approximation. Fuzzy logic can be amalgamated with data mining so as to diminish amounts of the input data as well as to choose occurrences which expose traffic anomalies (Atre, & Singh, 2016). The fuzzy logic technique is extremely effective for purposes of identifying innovative and new network attacks. A fuzzy logic methodology studies and evaluates the frequency of activities, CPU activity and device session connection durations. Fuzzy systems have the ability to dynamically consolidate data inputs fed from variable devices in the network. Fuzzy logic techniques enable quick construction of “if-then” criteria that can mimic and recognize unauthorized intrusions (Mkuzangwe, & Nelwamondo, 2017).

q. Genetic Algorithms Technique

Genetic algorithms methodology has been used in computational biology by naturally selecting and evaluating the evolution of domains. The Genetic Algorithms initiates an indiscriminate generation of an enormous number of potential applications. The accuracy and effective results generated by each domain enable classification and ranking. The better-ranked domain applications edge out moribund and programs deemed less effective. Sequentially high performing applications outlast those with low accuracy, ultimately only the strong survive (Bhattacharjee, Fujail, & Begum, 2017).

Conceptual Framework

An anomaly-based intrusion detection model that fuses SVM and ANN is thus proposed to address the gap between bottlenecks in the two machine learning techniques. This is achieved by combining a variable matrix of the two machine learning techniques. A fusion of artificial neural networks and support vector machine data classifier was implemented. This enables proper monitoring and profiling traffic emanating from the WSN. The neural network is also supported through reinforcement learning so as to maximize the cumulative result. The network is then trained by introducing internet packet traces. The technical model will have the following components;

a) A Data mapping separator
    i) A Support Vector Machine.
b) Anomaly Detection Engine
    ii) An Artificial Neural Network
c) Alarm/Reporting Arm.

The Network collects all incoming/outgoing data transitioning through the interfaces. Packets are separated depending on interest and mapped accordingly to a higher dimensional feature space. This is fed into a Support Vector Machine that transforms a linearly non-separable problem into a linearly separable one. This is due to its strengths in data classification. Further, the classified data is fed into an artificial neural network that performs pattern recognition tasks. The ANN makes us of modified probabilistic radial basis function. Data packets with an anomalous symbol are thereafter passed into the anomaly detection engine. If the data is positive, an alarm is raised, and a particular anomaly is reported.

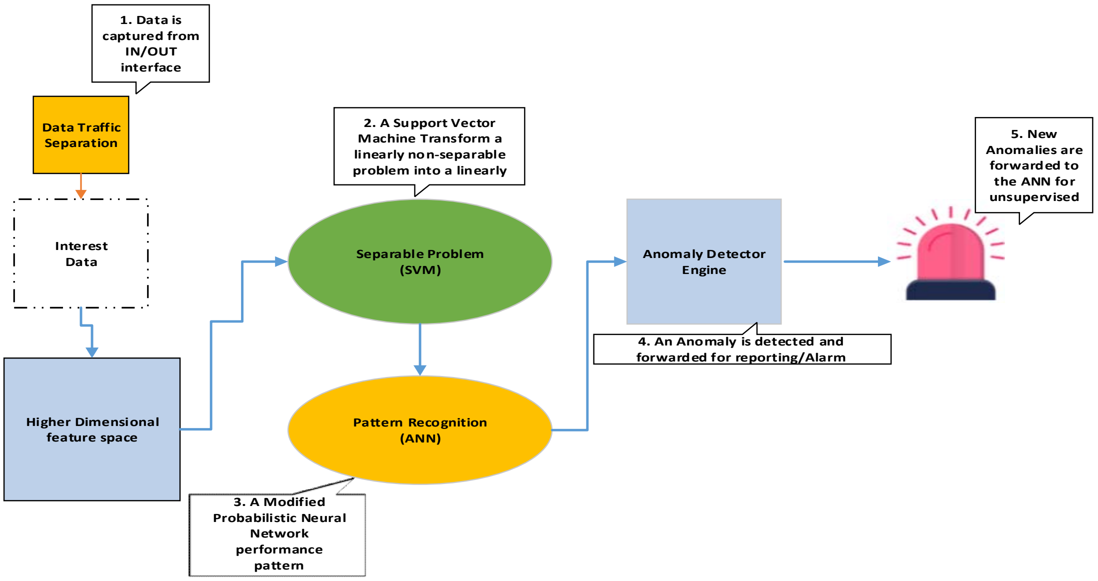

1. Data is captured from IN/OUT interface
Data Traffic Separation
Interest Data
Higher Dimensional feature space
2. A Support Vector Machine Transform a linearly non-separable problem into a linearly
Separable Problem (SVM)
Pattern Recognition (ANN)
3. A Modified Probabilistic Neural Network performance pattern
Anomaly Detector Engine
4. An Anomaly is detected and forwarded for reporting/Alarm
5. New Anomalies are forwarded to the ANN for unsupervised

Figure 2: Conceptual Framework

METHODOLOGY

Implementation of the Fused Machine Learning Intrusion Detection Model.

Simulation on NS 2 on Linux

This section presents the methodology used to implement the model on NS 2 on Linux. The purpose of implementing this study through a Linux simulation was to as to provide a quick, cost-effective virtual experimentation of the fused machine learning intrusion detection model. This virtual implementation of the real experiment enabled predictions about imminent future behaviours of the IDS possible quickly and cost-effectively. This enabled assumptions and approximations on how the real experiment would perform accurately by drawing inferences concerning the operating characteristics of the real IDS system.

The implementation of the ABIDS Model in a MANET environment was performed as follows:

i) The MANET network was simulated in NS 2 on Linux operating system.
ii) Descriptive script for the fused intrusion detection system was thereby introduced within the MANET network.
iii) The simulation was analyzed and simulated to determine if the MANET network while vulnerable to attacks like replay, relay, and man-in-the-middle attacks is actually experiencing the same.
iv) The Simulation was limited to a period of 5 hours since the MANET network can propagate data to infinity if no attack is discovered; or at least until the computer runs out of memory.

Implementation Smart Healthcare Network on Raspberry Board Microcontroller

For purposes of implementing a live experiment of a MANET network, a testbed was set up to fashion a health monitoring device, i.e. – a smartwatch with the capability to collect blood pressure data. However, since the smartwatches are proprietary, there was need to reconfigure certain aspects of the MANET; thus, Raspberry Pi microcontrollers were introduced into the network. They were critical in the ability to upload enhanced software, thus to be able to propagate, this research created a testbed for the testing and propagation of patient body pulse levels. A Raspberry Pi Board was set up as a MANET network. The microcontrollers were then connected to a smartwatch, which would propagate data to and from the board.

A Raspberry Board microcontroller was used in the collection of patient health data that was propagated over the MANET. The Raspberry Pi is a low cost, high processing and with easy integration to a pulse sensor. This microprocessor uses an Atmel AVR processor. The Raspberry Pi is embedded with standard programming language compiler and firmware which will execute the software. The Raspberry microcontroller has 14 digital input/output pins, which can accommodate analogue input. The following items are required for this prototype;

a) Raspberry Pi 3 Model B+
b) Smart Watches
c) MANET Radio Modules
d) Character Display Module
e) 5Volts Voltage Regulator
f) 3 Micro Switches
g) Prototype Circuit Board
h) SD Memory card

Below is the Raspberry Pi Model B+ specification for this prototype;

- SoC: Broadcom BCM2837B0 quad-core A53 (ARMv8) 64-bit @ 1.4GHz
- GPU: Broadcom Videocore-IV
- RAM: 1GB LPDDR2 SDRAM
- Networking: Gigabit Ethernet (via USB channel), 2.4GHz and 5GHz 802.11b/g/n/ac Wi-Fi
- Bluetooth: Bluetooth 4.2, Bluetooth Low Energy (BLE)
- Storage: Micro-SD
- GPIO: 40-pin GPIO header, populated
- Ports: HDMI, 3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)
- Dimensions: 82mm x 56mm x 19.5mm, 50g

PROTOTYPE DEVELOPMENT AND RESULTS

This section presents the results of the objective of the study, which set out to implement the fused machine learning intrusion detection model for the provision of smart health care in MANETS. To assure the validity of the results, research triangulation was done using a simulated experiment and a live experiment. According to the National Academies of Sciences (2018), triangulation refers to the use of more than one method of collecting data on the same topic to assure the validity of the research.

Implementation of the MANET IDS on Linux using NS 2

In the simulation step, a MANET network was implemented on Linux, with the following set of objectives;

i) Set up a typical MANET network to depict mundane MANET disposition.
ii) Put the MANET network under various malicious attacks.
iii) Introduce the fused intrusion detection system to protect the MANET from these attacks.

In the proposed model, as shown in Figure 4 below, every node in the mobile ad-hoc network participates in intrusion detection. Each node is responsible for identifying signs of intrusion; however, neighbouring nodes can collaboratively investigate in a broader range. The MANET network was comprised of the following devices, as shown;

i) A total of 25 devices were configured
ii) Two (2) data source devices where data originated
iii) Two (2) data destination devices where data was sent to
iv) Two (2) attacker/malicious devices which propagated various attacks

The MANET network is designed and implemented on NS 2 as shown in figure 4 below;

Figure 3: MANET with RFDs, FFDs and Malicious radios.

There are 25 participating devices in the network, of which 19 are set to promiscuous mode but can go live when needed to undertake the MANET under stress tests. Two devices (marked green) are set to send/originate data which is received by two devices (marked blue). Within this ecosystem, there are two malicious nodes (marked red), which are the sources of various attacks – blackhole and DDOS. This IDS analyzes the packets going into or out of the MANET, in search of undesirable and suspicious activities. To effectively monitor and protect against threats, a machine learning module is added by creating a description of newly discovered abnormalities. The device(s) generally has to find a match between current activities and anomalies, only when a positive match is found, does the alarm is generated. This anomaly-based detection machine learning technique creates normal profiles of system states or user behaviours and compares them with current activities within the MANET. If a significant deviation is observed, the IDS raises the alarm and most importantly adds this to its learning gene. The Linux IDS Script for the MANET Topology is presented in appendix B, code listing 1. Line 16 of appendix B, code listing 1 below specifies the number of devices;

```
16.     set val(nn)     25                  ;# number of mobilenodes
```

While line 44 of appendix B, code listing 1 enables ad-hoc routing for MANETS.

```
44.     $ns node-config -adhocRouting  $val(rp) \
```

As a result, a MANET environment is created, and data is propagated on the network simulating a real-world MANET environment.

a. Simulation of Machine Learning Phase 1- Setting Algorithm learning parameters for normal and abnormal modes

Figure 5 below shows the MANET in normal propagation mode.

Figure 4: Normal Data Propagation Mode

Figure 6 below shows the MANET propagation when malicious packets are sent;

Figure 5: MANET With Malicious Data is Propagating

Since in a MANET, mobility-induced dynamics make it challenging to distinguish between normalcy and anomaly, there is need to have a good representation of normal devices (in promiscuous mode) that can be used to comparatively give what is considered as the following

i) Normal Idle
ii) Normal Propagating (sending)
iii) Normal Propagating (receiving)

As well as;

iv) Anomalous Idle
v) Anomalous Propagating

This proposed IDS provides a promising alternative and specification based on detection techniques that combine the advantages of anomaly detection and misuse detection by using machine studied specifications to characterize legitimate system behaviours.

The support vector machine code for collecting interesting data into the engine is listed under appendix B, code listing 2.

Line 65 of code listing 2 shows the importation of data for purposes of classification, as shown below.

```
set list {}
                    foreach dir $import_dirs_ {
                            lappend list [$self file join $dir \
[$self class_to_file \
```

This enables interest data to be loaded for malicious detection. Attacks are identified as deviations from a normal profile and is improved by continuously comparing propagation of the 25 nodes within the MANET. However, the downside is that the development of detailed specifications can be time-consuming. The Linux IDS Script for Learning Normal and Anomalous activity for machine learning phase 1 is presented in appendix B, source code 3. Line 29 of code listing 3 creates a tracefile that records all anomalous activity relating to blackhole attacks. This enables the IDS to learn how blackhole attacks are propagated.

```
29.     set tracefile [open blackhole.tr w]
```

The resultant file is blackhole.tr. A Trace file is written by an application to store overall network information. The trace file should contain topology information, e.g., nodes, links, as well as packet traces. Line 32 of code listing 3, initiated visual reports to enable viewing of the propagation of packets during a black hole attack.

```
32.     set namfile [open blackhole.nam w]
```

To differentiate between the various devices on the network, colour coding was implemented. The red colour would indicate a malicious attacker; green would indicate the source of data and blue the destination as shown on lines 201 to 211 on code listing 3, appendix B.

```
201.    $ns at 0.0 "$n13 color red"
202.    $ns at 0.0 "$n13 label Attacker"

204.    $ns at 0.0 "$n23 color green"
205.    $ns at 0.0 "$n23 label Source"

207.    $ns at 0.0 "$n21 color green"
208.    $ns at 0.0 "$n21 label Source"

210.    $ns at 0.0 "$n29 color blue"
211.    $ns at 0.0 "$n29 label Destination"
```

Nam is a TCL based animation tool that enables viewing of network simulation traces and real-world packet traces. It supports topology layout, packet-level animation, and various other data inspection tools.

b. Machine Learning Phase 2 - Propagating Blackhole Attacks on a MANET using NS2

During a BlackHole MANET Attack, it is critical to ensure RREP is set with a Destination address and organization more noteworthy than the destination arrangement of the receiver node. This makes the sender node trust the black hole node, and addition interconnects with the malicious blackhole node in its place of the real trusted destination node. This mischievous setting frequently harms the victim node interfacing with the attacker and thus consuming all network resources rendering assets not only unusable but also causes packet loss. Figure 7 below illustrates a blackhole attack topology.

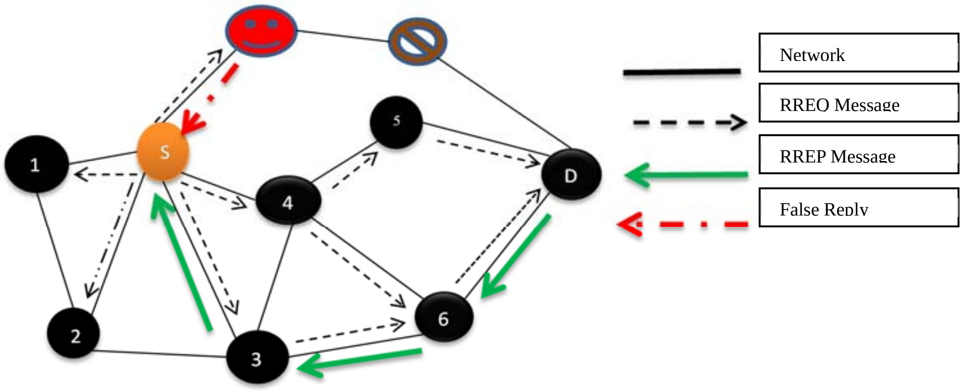


Figure 6: Topology of a black hole attack.

Figure 7: Implementation of IDS script for a blackhole learning environment

In this study, blackhole attacks were propagated by malicious node 5 within the MANET. To simplify the attack, 7 nodes were used for purposes of initial assisted machine learning with the following functions;

i) Node 1, 2, 4 and 6 – Normal promiscuous mode.
ii) Node 0 – Source of Data
iii) Node 6 – Destination of the Data

This design and implementation is as indicated in figure 8 as shown above;

In this learning phase, Node 5 propagates malicious blackhole scripts that forces Source Node (0), to redirect traffic meant for destination Node (3) to route traffic to malicious Node (5) instead. During this machine learning phase, behavioural genetics of an idle node, source node, destination node and malicious node is reviewed and logged. The Linux IDS Script for Propagating a Blackhole attack on the MANET Topology in figure 8 is presented in appendix B, code listing 3. Line 29 of code listing 3 creates a tracefile that records all anomalous activity relating to blackhole attacks. This enables the IDS to learn how blackhole attacks are propagated.

29. set tracefile [open blackhole.tr w]

The resultant file is blackhole.tr. A Trace file is written by an application to store overall network information. The trace file should contain topology information, e.g., nodes, links, as well as packet traces. Line 32 of code listing 3, initiated visual reports to view the propagation of packets during a black hole attack, by colour coding.

32. set namfile [open blackhole.nam w]

Nam is a TCL based animation tool that enables viewing of network simulation traces and real-world packet traces. It supports topology layout, packet-level animation, and various other data inspection tools.

c. Implementation and Testing of the MANET IDS on Linux and NS 2 against TCP SYN Flood, Blackhole and malicious traffic

After integration of the algorithms simulated, it was critical to evaluate the performance of the integrated machine learnt IDS against TCP SYN Flood, Blackhole and malicious traffic on the MANET. The MANET network was set-up with the same exact Nodes, namely;

i) Two (2) source nodes to send data.
ii) Two (2) destination node to receive data.
iii) Two (2) attacker/malicious devices (blackhole, TCP SYN).
iv) In total, 30 devices converged within the network.

The MANET topology for this network was designed and implemented, as shown below by figure 9.

Figure 8: MANET topology with IDS algorithm injected into the FFD

To achieve this, the following are important member functions that participate in the project for the following key objectives;

- Member function to specify and initiate the packet format.
- Member function to create a scheduler of events and actions within the MANET.
- Member function to enable selection of the default packet addressing scheme. This can be IP or IPv6.
- Member function to create MANET devices such as nodes and links.
- Member function to interconnect network component objects created, e.g. via Bluetooth, AODV etc.
- Member function to create connections between agents. These can be either TCP or UDP connections.

All the above member functions are integrated into a file ns-lib.tcl which performs the integration. The code for this file is found under appendix B, code listing 4. This file is, however, truncated for editing purposes. Key highlights are as follows;

a. $ns trace-all file-des – traces and records all simulation events
b. proc finish {} – terminates the simulation
c. set n[0] [$ns node] – sets up the nodes (in this case 30 nodes)
d. $ns duplex-link nodex nodey bandwidth – kicks off a duplex communication
e. set tcp [new Agent/TCP] – enables TCP communication on object instances
f. $ns attach-agent – is a member function that matches traffic to objects
g. $ns connect – establishes a virtual logical connection between two or more objects.

The Script for the fused IDS Model for MANET Topology in Figure 9 is presented in appendix B, code listing 3. There are important additions to the file ns-lib.tcl as indicated below;

Line 128 to 134 enables data logged by the member function **$ns trace-all file-des** to be fed into the SVM machine for purposes of classification. The populates and pipes the results into the port_file, if it does not exist, one is created.

```
128.set flag 1
129.} elseif {[file exists $PORT_FILE_] && [file readable $PORT_FILE_]} {
130.for kernel in kernels:
131.  svc = svm.SVC(kernel=kernel).fit(X, y)
```

```
132.  plotSVC('kernel=' + str(kernel))
133.} else {
134.set flag 1
```

Further, an addition to the IDS script enabled pattern recognition on the interest data. This was done by integrating an artificial neural network technique for purposes of performing pattern recognition as shown in line 184 to 189 of code snippet 4, appendix B

```
184.  INT n,i,j;
185.  for (n=0; n<NUM_DATA; n++) {
186.    for (i=0; i<Y; i++) {
187.      for (j=0; j<X; j++) {
188.        Input[n][i*X+j] = (Pattern[n][i][j] == 'O') ? HI : LO;
189.      }
```

In addition, any resultant data reviewed for pattern recognition that was a true positive was forwarded to the alarm as shown in line 635 to 638 of code snippet 4, appendix b

```
635.$self notifyObservers $now
636.$self instvar netAddress
637.if ![info exists netAddress] {
638.set netAddress [new Address]
```

Implementation of MANET IDS prototype on Raspberry Pi and Generic Smart-watch

This section presents the findings of the implementation of the MANET anomaly-based intrusion detection model for smart health care prototype using a live experiment via a Proof of Concept methodology. The project prototype implementation involved setting up of a live MANET with a smart-watch with blood pressure monitoring capability, which forwarded this data to a fully functioning device - Bluetooth Router configured on a Raspberry Pi.

The prototype was implemented to measure and monitor blood pressure and forward the same readings through the MANET ecosystem. Blood pressure measurements were carried out through smart-watches wrapped on the wrist. The generic smart-watch was used to collect data in the form of blood pressure and propagated this data to the fully functional device (or Bluetooth router).

The fused IDS was installed on the Raspberry Pi and dynamically monitored the system and users' actions in the system so as to detect intrusions. As the results show, the model successfully protected against blackhole attacks, brute force and TCP SYN based denial of service among other DDOS attacks.

The image below shows the Bluetooth smart-watch in Figure 10 and 11 that collected and forwarded client data.

Figure 9: Smart Watch Physical Address   **Figure 10: Smart Watch Collecting Readings**

Figure 10 above shows the Bluetooth smart-watch with a physical MAC address of A4:C1:3C:EB:07:DB as the address which propagated data to the router. Figure 11 above shows the smart-watch initiating reading of blood pressure on a user.

The generic smart-watch acted as the reduced function device on the edge of the network. The smart-watch collected data in the form of blood pressure and propagated this data to the fully functional device (or Bluetooth router). The smart-watch has the following specifications;

a) Processor - Nordic-Nrf51822
b) Operating System - Proprietary OS
c) Display - 0.86 inch OLED
d) RAM –
e) Battery - 60 mAh Polymer lithium battery Normal use:7days;Standby -15days
f) Sensors - Pedometer, Heart Rate Monitor, 6 axis acceleration sensor, Blood Pressure Measurement
g) Input voltage: 5V
h) Weight: 6.9 grams net weight
i) Heart rate: S7000
j) BT: BT 4.0 /compatible (android and IOS)
k) Touch: touch mode button
l) Bluetooth- 4.0BLE forward compatible (Low power consumption)

The client propagates data via Bluetooth 4.0 but is forward compatible with Bluetooth 4.5 as well.

Figure 12 below shows the Raspberry Pi, which acted as the Bluetooth MANET fully functional device that collected and forwarded client data from the MANET clients for purposes of evaluation on the IDS system. The IDS system was uploaded on the single board computing device since the smart-watches run a proprietary operating system. This is the newest device as at the research date, which runs on a 1.4GHz 64-bit quad-core processor, dual-band wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and Power-over-Ethernet support (with separate PoE HAT). Figure 22 below shows the Raspberry Pi device used in the setup.

Figure 11: Raspberry Pi 3Model B+ (Source: Buy a Raspberry Pi 3 Model B – Raspberry Pi. (n.d.). Retrieved from https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/)

The Raspberry Pi 3 Model B+ has the following specifications;

a) SOC: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC.
b) CPU: 1.4GHz 64-bit quad-core ARM Cortex-A53 CPU.
c) RAM: 1GB LPDDR2 SDRAM.
d) WIFI: Dual-band 802.11ac wireless LAN (2.4GHz and 5GHz) and Bluetooth 4.2.
e) Ethernet: Gigabit Ethernet over USB 2.0 (max 300 Mbps). Power-over-Ethernet support (with separate PoE HAT). Improved PXE network and USB mass-storage booting.
f) Thermal management
g) Video: Yes – VideoCore IV 3D. Full-size HDMI.
h) Audio: Yes.

i) USB 2.0: 4 ports
j) GPIO: 40-pin
k) Power: 5V/2.5A DC power input
l) Operating system support: Linux and Unix

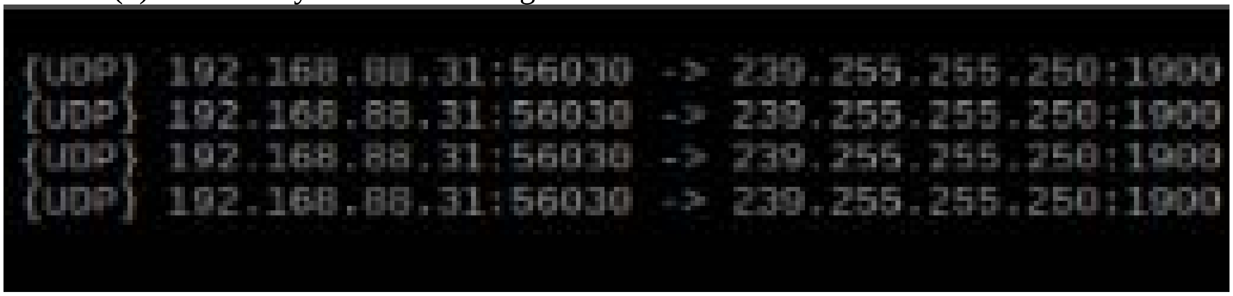
The figure below shows the 16Gb micro SD card on which the applications, scripts and IDS were pre-installed so as to run on the Raspberry that doesn't have enough storage.

a. Equipment Configuration, Setup and Data Propagation

The python code in Appendix C code listing 1, connects to the smartwatches which were used in this experiment to form a MANET network. The python modules needed to run the IDS are first imported in line 1 to line 4 of appendix C, code listing 1. The imported modules include the subprocess module to run the hciconfig tool and the gat tool as well as the threading module, which allows multiple connections. The code scans for BlueTooth devices for ten seconds, then connects to the found devices and turns the blood pressure rate notifications on to get notified when the blood pressure changes. The IDS mainly uses Linux 'hciconfig' to search for BlueTooth devices and gat tool to connect and interact with them. This is shown in the code snippet below.

1. ***import subprocess***
2. ***from subprocess import ****
3. ***import time***
4. ***import threading***

As a result of this setup, the data being propagated on the MANET is visible when scanned, as shown in Figure 13 below. The captured Raspberry Pi interface depicts a normal UDP communication between source address 192.168.88.31 forwarding packets to 239.255.255.250; the MANET network is propagating data in a normal case scenario as depicted in the Finite State Machine (1) without any attack emanating within the network.



```
{UDP} 192.168.88.31:56030 -> 239.255.255.250:1900
{UDP} 192.168.88.31:56030 -> 239.255.255.250:1900
{UDP} 192.168.88.31:56030 -> 239.255.255.250:1900
{UDP} 192.168.88.31:56030 -> 239.255.255.250:1900
```

Figure 12: Snort analyser sniffing packets on the MANET

Figure 13 above shows normal packet flow within the MANET using Snort. Snort is a free and open-source network analysis module that can perform real-time traffic analysis and packet logging on Internet Protocol (IP) networks. Snort performs protocol analysis, content searching and matching. In this case, the home address of 192.168.88.31 was captured with a special analysis on port 56030 for both reliable TCP and unreliable UDP packets on the MANET.

b. Python Code to Capture Usage, RAM and Network Bandwidth on the MANET Topology

This section presents the results of monitoring device activity as a contributor to anomalous activity. A MANET node undergoing heavy attack can receive tonnes of requests which, as a consequence, causes the device to overuse its resources – CPU, RAM and bandwidth. These activities on the MANET device itself can be monitored to establish where a device is busy or is overusing its resources in response to an attack. The code lines 13 to 33 on code listing 2, appendix C was written in Python to capture the amount of data propagated between the devices, RAM, CPU usage as well as bandwidth and throughput on the network. Special attention is brought to line 21 and 22 specifically monitor the amounts of packets uploaded and downloaded respectively. This is shown in the code snippet below.

```
21.     upload=psutil.net_io_counters(pernic=True)[network_interface][0]
22.     download=psutil.net_io_counters(pernic=True)[network_interface][1]
```

This activity is thereafter evaluated against what is considered normal and/or anomalous so as ascertain whether the activity is potentially malicious are outright malicious.

The outcome of this section was data recordings on system and process utilities usage by the MANET device. The logged file contained data showing CPU, RAM and Bandwidth usage which can be monitored to identify a pattern of high resource usage that is tantamount to a node undergoing heavy attack.

c. Machine Learning using Support Vector Machines: Data importation capture and Separation.

This data importation capture and separation module captures general packets and segregates interest data by mapping each type and address accordingly to a higher dimensional feature space. This output is consequently taken as the source input for the Support Vector Machine for separation.

To achieve data capture, snort – which has capabilities for sniffing and packet logging was used to capture packets from a manipulated syn flood attack. Figure 14 below shows the MANET under DDoS attack, to which the snort was able to detect the anomaly as being a possible TCP DoS attack. This was effected by running LOIC (Low Orbit Ion Canon). LOIC is a free network analysis tool, which is popular for initiating DOS attacks. LOIC tool is freely available on the Internet. LOIC performed the DDoS attack by sending successive SYN requests to addresses on the MANET in an attempt to rid the devices of resources, thus making it unresponsive to legitimate requests. Figure 14 shows DoS recognition on the MANET.

Figure 13: MANET under manipulated DDOS attack

Figure 14 above receives numerous fictitious requests from a public IP 45.44.244.222 attacking various open and accessible ports in the range of 49152-65535 towards 192.168.88.31 on its open http port 80. Most DDoS tools listen on the dynamic range of 49152-65535 ports to find available and/or unprotected ports.

Based on combining misuse with anomaly detection, our IDS for MANETs accurately and efficiently detects attacks such as DoS, replay attack and compromised nodes. Our results have shown a great promise for the future, which would focus on making the scheme more robust by taking a broader range of attacks into consideration and making use of audit data to adjust the threshold accurately.

This algorithm provides the ability to perform unsupervised neighbours-based learning techniques. Unsupervised nearest neighbours provide a foundation for various other learning methods. Two techniques are used in particular - manifold learning and spectral clustering. In this particular module data captured from the MANET module is classified by the KNeighborsClassifier and uses the following methods;

a) Fit (X, y) Uses X as training data and y as target values.
b) get_params ([deep]) Sets the various constraints.
c) Kneighbors ([X, n_neighbors, return_distance]) this calculates the K-neighbors of a point.
d) predict_proba (X) Computes all the various possibilities for the test data X.
e) score (X, y[, sample_weight])Computes possible variation from mean accuracy to avoid false alarm

The data logged from the capturing activity can be seen as Data Listing 1 and 2, appendix C. The python code in appendix C code listing 3, presents the integration of SVM algorithm for data importation capture and separation. This is shown in the code snippet below.

```
#loads the data set
data=[]
fp=open('all_logs','r')
for line in fp.readlines():
vals = line.strip().split(' ')
try:
l_elem =  [ float(i) if '.' in i else str(i) for i in vals ]
data.append(l_elem)
except Exception as e:
pass
import pandas as pd
import numpy as np
_dataset=np.asarray(data)
```

The python code in appendix C code listing 3, presents the scatter plot generation from the captured data. This is shown in the code snippet below.

```
plt.scatter(_dataset[:2591, 0], _dataset[:2591, 2], c='r', label='default')
plt.scatter(_dataset[2591+1:2591+2201, 0], _dataset[2591+1:2591+2201, 2], c='g',label='ddos')
plt.scatter(_dataset[2591+2201+1:2591+2201+2155, 0], _dataset[2591+2201+1:2591+2201+2155, 2], c='b',label='no_IDS')
plt.scatter(_dataset[2591+2201+2155+1+1012:, 0], _dataset[2591+2201+2155+1+1012:, 1], c='yellow',label='norm')
```

d. Machine Learning Implementation: Support Vector Machine for separable problems.

Support Vector Machine algorithms were preferred in this study due to their ability to give very high accuracy in comparison to other classifiers. While logistic regression and decision trees have been implemented before, SVMs were deemed perfect for intrusion detection on the basis of accuracy. The SVM data classifier segregates packets using a hyperplane with the largest amount of margin.

The model performs the following steps to achieve learning from the propagated data;

i) Prepare data:

   This is performed prior by SVM so as the right and relevant interest data is captured, segregated and analyzed.

ii) Create an instance of a Linear SVM classifier:

   Generate hyperplanes which segregate the interest data according to behaviour, rate and type separating the two classes correctly.

iii) Train a Linear SVM classifier:

   This compares the two vectors separated by the decision boundary or hyperplane with the two nearest neighbour packets data points (D+ and D-).

Thereafter, the interest data is input into the training stage so as to enhance the IDS engine's accuracy. This is shown on the code line 25 to 27 from code listing 4, appendix C, where the engine is trained on the behaviours.

```
25.     train_data,test_data,train_label,test_label = train_test_split(dataset.iloc[:,:1], dataset.iloc[:,2], test_size=0.2, random_state=1)
26.     #the k (n_neighbors) parameter is often an odd number to avoid ties in the voting scores. eg 1-9 neighbors = np.arange(1,9)--has 9 neighbors
27.     #2 numpy zero matrices namely train_accuracy and test_accuracy each for training and testing accuracy
```

SVM achieves training and testing set split by importing sklearn library which has an in-built splitting function called train_test_split. This uses the random_state as a seed that takes a random_state as input. Changing the number of seeds will also affect and change the split of the data. Maintaining the same random_state while running the cell multiple times will ensure the data splitting remains unchanged.

Further, the performance of the training and testing are presented by running the code lines 36 to 39 from code listing 4, appendix C

```
36.     knn = KNeighborsClassifier(n_neighbors=11)
37.     knn.fit(train_data, train_label)
38.     train_accuracy[i] = knn.score(train_data, train_label)
39.     test_accuracy[i] = knn.score(test_data, test_label)
```

The output of this section is displayed in the following Figure 15 below, which plots the results and correlation between TCP and packet propagation in the MANET. The results indicated here shows that the Fused Model successfully separated interest data as required. The results in Figure 15 also shows how the SVM separated the data packets propagated on the network using colour codes where green shows data from the source device, blue shows data from the destination device and red data from a malicious device.

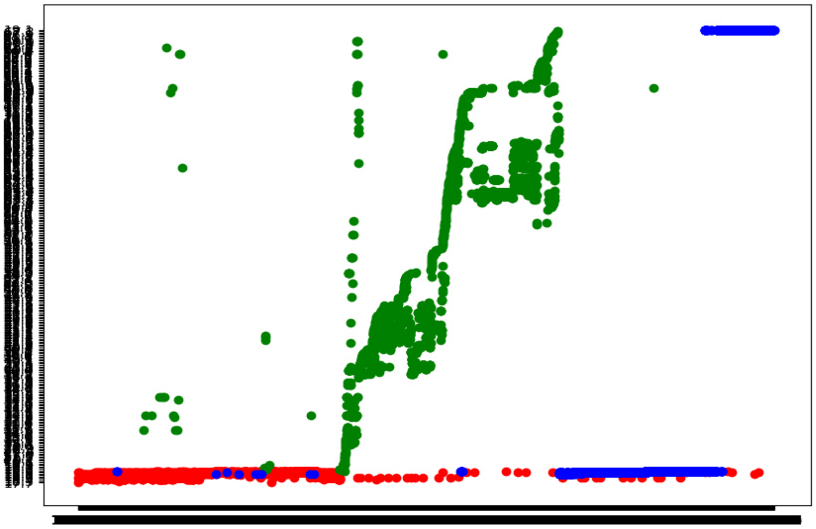


Figure 15: Data propagated in the MANET correlation

The accuracy of the interest data was evaluated further by feeding it into an artificial neural network, and MLP Classifier initiated to perform train/test split and results compared. The results of this test are presented in Figure 16 below that shows the Artificial Neural Network test split. It plots the training and testing accuracy, with accuracy against a varying number of neighbour's graph. The results below indicate that the data inherited from the support vector machine (SVM) had negligible deviation. The difference in results between training data and test data is negligible.

The k value from this graph indicates that the fused anomaly model performs the best at the hyperplane between D+ of 0.3 to D- of 04. The difference between test data and training data is presented in Figure 16

.

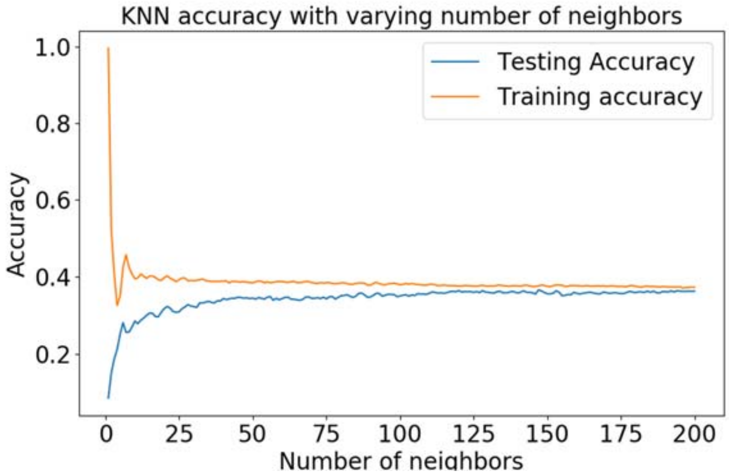


Figure 16: KNN accuracy of Testing and Training Data

e. Machine Learning Fusion: Support Vector Machine into Artificial Neural Networks

This section involved integrating two machine learning concepts, each contributing to the strength of the model.

i. The Support Vector Machine, due to its strengths in data classification, identifies interest data and separates the data accordingly.
ii. This classified data is fed into an artificial neural network that performs pattern recognition tasks.
iii. The pattern recognition is performed on interest data on both sides of the hyperplane, in accordance with the anomaly data pattern.
iv. Data packets with the anomalous symbol are thereafter passed into the anomaly detection engine.

The resulting positive vector is imported into the Artificial Neural Network algorithms MLPClassifier and hidden_layer_sizes so as to perform function approximation quickly. The ANN module essentially performs three major tasks which include pre-processing already performed by SVM;

a) Train Test Split – This is achieved by dividing the achieved vector data to both training and test splits. Training will be achieved by using the training data, and performance of the system tested through the test data.
b) Feature Scaling – This will aid to review how the ANN makes predictions when a large number of packets, which are anticipated if and when the MANET is propagating huge amounts of packets or when the MANET is experiencing high data rates while maintaining accuracy.
c) Alarm - The resulting positive match to anomaly patterns is identified and forwarded for reporting.
d) Learning Loop - Newly found anomalies that have not been experienced but do not satisfy norms are filtered, clustered and forwarded to the ANN for learning and future reference.

The code listing 4, appendix C shows the implementation of train test, feature scaling, alarm and learning for ANN in python. This is achieved by importing the class perceptron from scikit and creating a new perceptron to handle the data x and y. This is shown in the truncated code listing 5; Appendix C. A snippet is displayed below.

```
Perceptron(alpha=0.005, class_weight=None, early_stopping=False, eta0=0.1,
      fit_intercept=True, max_iter=30, n_iter=None, n_iter_no_change=2,
      n_jobs=None, penalty=None, random_state=42, shuffle=True, tol=0.001,
      validation_fraction=0.02, warm_start=False
```

Thereafter, the MPL Classifier is imported into the IDS. A multilayer perceptron (MLP) is a feedforward artificial neural network inbuilt function, that maps various sets of input data onto a set of appropriate outputs. This is presented in code listing 5, appendix C and also in the lines 7 to 12 below.

```
7.    from sklearn.neural_network import MLPClassifier
8.    X = [[0., 0.], [0., 1.], [1., 0.], [1., 1.]]
9.    y = [0, 0, 0, 1]
```

```
clf = MLPClassifier(solver='lbfgs', alpha=1e-5,
                    hidden_layer_sizes=(5, 2), random_state=1)
print(clf.fit(X, y))
```

To check on the accuracy of the module results, a function to calculate the number of neurons is activated. The purpose is to count the number of neurons between the inputs and the outputs of the ANN. For best accuracy, they should be equal to the two-thirds of the sum.

```
for i in range(len(clf.coefs_)):
    number_neurons_in_layer = clf.coefs_[i].shape[1]
    for j in range(number_neurons_in_layer):
```

The resulting IDS is introduced in the MANET, and a DoS is executed to test its ability. Figure 17 below shows the introduction of the smart IDS to counter DDOS attacks within the MANET; the results presented here show that is able to detect anomalous events within the MANET.

Figure 14: IDS introduced into the MANET detects anomalous activity (DDOS)

Figure 17 above shows deployment of the IDS causes an alarm to go off after positively identifying DDOS attacks on the MANET and consequent successful segregation of the same.

f. The Fusion of ML techniques into the IDS

This section presents the integration of machine learning techniques into the IDS as envisaged in the model design. Figure 18 shows the resulting framework where the new machine learning elements are fused into the existing snort IDS rules.

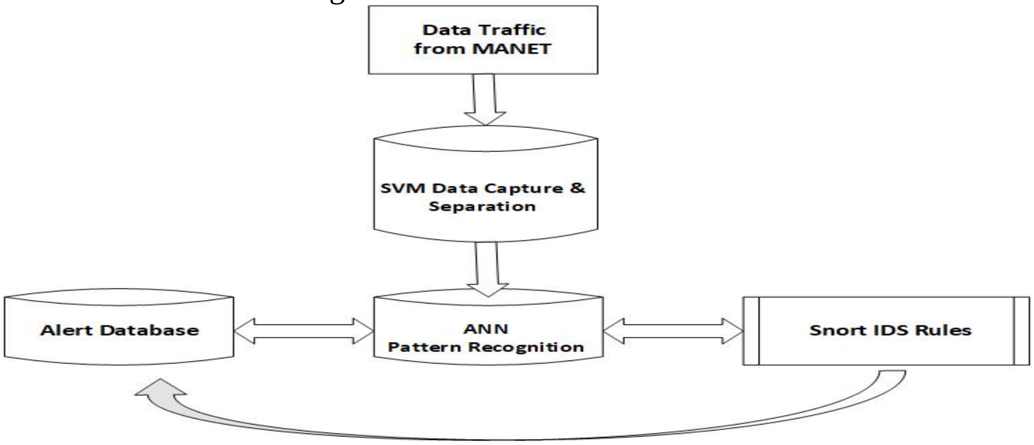


Figure 18: IDS Fusion Framework

The following code files are added to original IDS module:

- src/preprocessors/MLP_ANN.c
- src/preprocessors/MLP_ANN.h
  This are MultiLayer Perceptron (MLP) Perceptron neural network source files.
- src/preprocessors/svm_data.c
- src/preprocessors/svm_log.h

This is a preprocessor source file that integrates the Support Vector Machine SVM data capture and separation techniques.

The following files are edited on the original IDS:

- etc/snort.conf
- src/plugbase.c
- src/preprocessors/flow/flow_callback.c

This integration in detail within the files is indicated below

g. etc/snort.conf

The following values on snort.conf were changed so as to enable portscan to use the machine learning SVM techniques for data collection and separation. The source code is found under Appendix D Data Listing 1 from line 100 to 105

```
100 preprocessor portscansvm: ignorebc 1 \
 101 analyze_thr_lower 100 \
 102 analyze_thr_upper 1600 \
 103 sense_level 0.05 \
 104 net_topology 0 \
 105 log_method 1
```

h. src/plugbase.c

The following values on plugbase.c were added so as to include the machine learning source files added – svm_log.h and MLP_ANN.h. These files invoke a multilayer perceptron which is a class of feedforward artificial neural network that calculates and produces values from a set of given inputs. The source code is found under Appendix D Data Listing 2 at various lines 54, 66, 154 and 155.

```

54 #include "preprocessors/svm_log.h"
66 #include "preprocessors/MLP_ANN.h"
154 extern PreprocConfigFuncNode *preproc_svm_log;
155 extern PreprocConfigFuncNode *preproc_MLP_ANN;
```

i. src/preprocessors/flow/flow_callback.c

The following values on flow_callback.c were added so as to include the svm and ANN source files added – svm_data.c and MLP_ANN.c. The source code is found under Appendix D Data Listing 3 at lines 265 and 267.

```
265  src/preprocessors/Stream6/svm_data.c,
267  src/preprocessors/Stream6/MLP_ANN.c,
```

The result of this section is a Fused IDS engine which contains the following innovative attributes;

i) A preprocessor which separates data using support vector machines as opposed to the original data preprocessor
ii) An Artificial Neural Network that performs pattern recognition from received portscans captured and separated having common characteristics of anomalous nature.
iii) Captured and stored dataset of normal and anomalous traffic
iv) Data sets that have been learned by the ANN during the training time

v) Weights that are defined by the simulator using ANN learning function
vi) A fused IDS that drop packets that meet the anomalous criteria described by snort rules but earned and identified by machine learning techniques.

CONCLUSION

This study implemented the model through project research triangulation. This involved setting up two experiment scenarios, in using a simulated experiment and a live prototype experiment. The virtual simulation was achieved using NS2 in Linux, and the live experiment was achieved by using a generic smartwatch and raspberry pi for purposes of setting up the dummy MANET for smart healthcare. Both setups experienced an induced DDoS attack that further worked to prove the MANETs weaknesses. In the experiments above, a fused IDS was also implemented so as to review the ability to protect the MANET successfully.

The implementation involved editing open-source intrusion detection system based on snort rules and infusing two attributes of machine learning. This was achieved by introducing following code files are added to the original IDS module; namely the src/preprocessors/MLP_ANN.c and src/preprocessors/svm_data.c. These are MultiLayer Perceptron (MLP) Perceptron neural network source files as well as the src/preprocessors/svm_log.h and src/preprocessors/MLP_ANN.h header files. This is a preprocessor source file that integrates the Support Vector Machine SVM data capture and separation techniques. By configuring the snort.conf file and flow_callback.c file, these extra files were integrated into the Fused IDS.

REFERENCES

Agrawal, S., & Vieira, D. (2013). A survey on Internet of Things. *Abakós, 1*(2), 78-95.

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials, 17*(4), 2347-2376.

Almusallam, N. Y., Tari, Z., Bertok, P., & Zomaya, A. Y. (2017). Dimensionality reduction for Intrusion Detection Systems in multi-data streams: A review and proposal of unsupervised feature selection scheme. In *Emergent Computation* (pp. 467-487). Springer International Publishing.

Aguiar, RL, S. Sargento, A. Banchs, CJ Bernardo, M. Calderon, I. Soto, M. Liebsch, T. Melia, & P. Pacyna. (2006, June). Scalable QoS-aware mobility for future mobile operators; *COM Magazine, 44*(6), 95-102.

Atre, A., & Singh, R. (2016). A Concept on Intrusion Detection System genetic algorithm, fuzzy logic and challenges: A review. *International Journal of Scientific Research in Science, Engineering and Technology, 2*(1), 287-89.

Bartoli, A., Dohler, M., Hernández-Serrano, J., Kountouris, A., & Barthel, D. (2011). Low-power low-rate goes long-range: The case for secure and cooperative machine-to-machine communications. In *Networking 2011 Workshops (pp. 219-230). Springer Berlin/Heidelberg.*

Bedi, G., Venayagamoorthy, G. K., & Singh, R. (2016). Internet of Things (MANET) sensors for smart home electric energy usage management. In *Information and Automation for Sustainability (ICIAfS), 2016 IEEE International Conference on* (pp. 1-6). IEEE

Bhattacharjee, P. S., Fujail, A. K. M., & Begum, S. A. (2017). Intrusion Detection System for NSL-KDD data set using vectorised fitness function in genetic algorithm. *Advances in Computational Sciences and Technology, 10*(2), 235-246.

Benkhelifa, E., Welsh, T., & Hamouda, W. (2018). A critical review of practices and challenges in intrusion detection systems for IoT: Toward universal and resilient systems. *IEEE Communications Surveys & Tutorials, 20*(4), 3496-3509.

Borgohain, T., Kumar, U., & Sanyal, S. (2015). Survey of security and privacy issues of Internet of Things. arXiv preprint arXiv:1501.02211.

Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cybersecurity intrusion detection. *IEEE Communications Surveys & Tutorials, 18*(2), 1153-1176.

Chaudhary, A., & Shrimal, G. (2019). Intrusion Detection System based on genetic algorithm for detection of distribution denial of service attacks in MANETs. Retrieved from SSRN 3351807.

Dhindsa, K. S., & Bhushan, B. (2019). Flow-based attack detection and defence scheme against DDoS attacks in cluster-based ad hoc networks. *International Journal of Advanced Networking and Applications, 10*(4), 3905-3910.

Folorunso, O., Ayo, F. E., & Babalola, Y. E. (2016). Ca-NIDS: A network intrusion detection system using combinatorial algorithm approach. Journal of Information Privacy and Security, 12(4), 181-196.

Gai, K., Qiu, M., Tao, L., & Zhu, Y. (2016). Intrusion detection techniques for mobile cloud computing in heterogeneous 5G. *Security and Communication Networks, 9*(16), 3049-3058.

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (MANET): A vision, architectural elements, and future directions. *Future Generation Computer Systems, 29*(7), 1645-1660.

Hui, K. L., Kim, S. H., & Wang, Q. H. (2017). Cybercrime deterrence and international legislation: Evidence from distributed denial of service attacks. *MIS Quarterly, 41*(2), 497.

Iyengar, A., Kundu, A., & Pallis, G. (2018). Healthcare informatics and privacy. *IEEE Internet Computing, 22*(2), 29-31.

Iqbal, I. M., & Calix, R. A. (2016, October). Analysis of a payload-based Network Intrusion Detection System using pattern recognition processors. In: *Collaboration Technologies and Systems (CTS), 2016 International Conference on* (pp. 398-403). IEEE.

Iqbal, S., Kiah, M. L. M., Dhaghighi, B., Hussain, M., Khan, S., Khan, M. K., & Choo, K. K. R. (2016). On cloud security attacks: A taxonomy and intrusion detection and prevention as a service. *Journal of Network and Computer Applications*, *74*, 98-120.

Jose, S., Malathi, D., Reddy, B., & Jayaseeli, D. (2018). A Survey on anomaly-based host Intrusion Detection System. In: *Journal of Physics: Conference Series 1000(1), 012049.* IOP Publishing.

Janis, P., Chia-Hao, Y. U., Doppler, K., Ribeiro, C., Wijting, C., Klaus, H. U. G. L., & Koivunen, V. (2009). Device-to-device communication underlaying cellular communications systems. *International Journal of Communications, Network and System Sciences, 2*(03), 169.

.

Kaur, M., & Saini, K. S. (2017). A Framework for recyclable household waste management system in smart home using MANET. In *Computing and Network Sustainability (pp. 213-223)*. Springer, Singapore.

Kabir, M. R., Onik, A. R., & Samad, T. (2017). A network intrusion detection framework based on Bayesian network using Wrapper approach. *International Journal of Computer Applications, 166*(4), 13-17

Kevric, J., Jukic, S., & Subasi, A. (2017). An effective combining classifier approach using tree algorithms for network intrusion detection. *Neural Computing and Applications*, *28*(1), 1051-1058.

Kenkre, P. S., Pai, A., & Colaco, L. (2015). Real-time intrusion detection and prevention system. In *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014* (pp. 405-411). Springer, Cham.

Khan, J. Y., Chen, D., & Hulin, O. (2014). Enabling technologies for effective deployment of Internet of Things (MANET) systems. *Australian Journal of Telecommunications and the Digital Economy, 2*(4), 1-22.

Kumar, G. R., Mangathayaru, N., & Narsimha, G. (2016). Intrusion detection: A text mining-based approach. *International Journal of Computer Science and Information Security, 14*, 76.

Kušen, E., & Strembeck, M. (2017). Security-related Research in Ubiquitous Computing--Results of a Systematic Literature Review. arXiv preprint arXiv:1701.00773.

Kolhe, P., Bhosale, S., Lathe, S., Mane, S., & Bhattad, R. (2016). Network intrusion detection by finding correlation between multiple features using K-means algorithm & multivariate correlation analysis. *Networking and Communication Engineering, 8*(3), 61-66.

Liu, M., Xue, Z., Xu, X., Zhong, C., & Chen, J. (2018). Host-based Intrusion Detection System with system calls: Review and future trends. *ACM Computing Surveys (CSUR), 51*(5), 98.

Le Dang, N., Le, D. N., & Le, V. T. (2016). A new multiple-pattern matching algorithm for the network intrusion detection system. *International Journal of Engineering and Technology, 8*(2), 94.

Marteau, P. F. (2019). Sequence covering for efficient host-based intrusion detection. *IEEE Transactions on Information Forensics and Security, 14*(4), 994-1006.

Mkuzangwe, N. N. P., & Nelwamondo, F. V. (2017, April). A fuzzy logic-based network intrusion detection system for predicting the TCP SYN flooding attack. In *Asian Conference on Intelligent Information and Database Systems* (pp. 14-22). Springer, Cham.

Moustafa, N., Creech, G., & Slay, J. (2017). Big data analytics for Intrusion Detection System: Statistical decision-making using finite Dirichlet mixture models. In *Data Analytics and Decision Support for Cybersecurity* (pp. 127-156). Springer, Cham.

Mir, N. M., Khan, S., Butt, M. A., & Zaman, M. (2016). An experimental evaluation of Bayesian classifiers applied to intrusion detection. *Indian Journal of Science and Technology, 9*(12), 1-7.

Michael, K. (2017). Go? Get chipped? A brief overview of non-medical implants between 1997-2013 (Part 1). *IEEE Technology and Society Magazine, 36*(3), 6-9.

National Academies of Sciences, Engineering, and Medicine. (2018). *Changing Sociocultural Dynamics and Implications for National Security: Proceedings of a Workshop*. Washington, DC: The National Academies Press. doi: https://doi.org/10.17226/25056.

Ngomane, I., Velempini, M., & Dlamini, S. V. (2018). The detection of the spectrum sensing data falsification attack in cognitive radio ad hoc networks. In *2018 Conference on Information Communications Technology and Society (ICTAS)* (pp. 1-5). IEEE.

Nigam, S., Asthana, S., & Gupta, P. (2016, February). MANET based intelligent billboard using data mining. In *Innovation and Challenges in Cyber Security (ICICCS-INBUSH), 2016 International Conference on* (pp. 107-110). IEEE.

Norman, D. (2017). Design, business models, and human-technology teamwork: As automation and artificial intelligence technologies develop, we need to think less about human-machine interfaces and more about human-machine teamwork. *Research-Technology Management, 60*(1), 26-30.

Reddy, G. D., Chutke, S., Reddy, M. S. V. R., & Rao, D. N. (2018). Wireless sensor network application for IoT based healthcare system. In *International Journal of Emerging Technologies and Innovative Research JETIR (Vol. 5, No. 2* (February-2018)). JETIR.

Rath, M., Swain, J., Pati, B., & Pattanayak, B. K. (2018). Network security: Attacks and control in MANET. In *Handbook of Research on Network Forensics and Analysis Techniques* (pp. 19-37). IGI Global.

Saha, H. N., Mandal, A., & Sinha, A. (2017, January). Recent trends in the Internet of Things. In *Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual* (pp. 1-4). IEEE.

Sundmaeker, H., Guillemin, P., Friess, P., & Woelfflé, S. (2010). Vision and challenges for realising the Internet of Things. *Cluster of European Research Projects on the Internet of Things, European Commission, 3*(3), 34-36.

Stankovic, J. A. (2014). Research directions for the internet of things. *IEEE Internet of Things Journal, 1*(1), 3-9.

Taher, K. A., Jisan, B. M. Y., & Rahman, M. M. (2019). Network intrusion detection using supervised machine learning technique with feature selection. In *2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)* (pp. 643-646). IEEE.

Roy, S. S., Mallik, A., Gulati, R., Obaidat, M. S., & Krishna, P. V. (2017, January). A deep learning-based artificial neural network approach for Intrusion detection. In *International Conference on Mathematics and Computing* (pp. 44-53). Springer, Singapore.

Saikumar, T., SudhaRani, G., Keerthi, K., Sneha, K., & Srikar, B. (2017). Modified improved Kernel Fuzzy adaptive threshold algorithm on modified level set method for picture segmentation. Evolution.

Wang, D., Long, Y., Xiao, Z., Xiang, Z., & Chen, W. (2016, July). A temporal self-organizing neural network for adaptive sub-sequence clustering and case studies. In *Computer, Information and Telecommunication Systems (CITS), 2016 International Conference on* (pp. 1-5). IEEE.

Yu, T., Sekar, V., Seshan, S., Agarwal, Y., & Xu, C. (2015). Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things. In *Proceedings of the 14th ACM Workshop on Hot Topics in Networks* (p. 5). ACM.

Vermesan, O., Friess, P., Guillemin, P., Gusmeroli, S., Sundmaeker, H., Bassi, A., & Doody, P. (2011). Internet of things strategic research roadmap. *Internet of Things-Global Technological and Societal Trends, 1*, 9-52.

.

Vongpradhip, S., & Rungraungsilp, S. (2012, January). QR code using invisible watermarking in frequency domain. In *ICT and Knowledge Engineering (ICT & Knowledge Engineering), 2011 9th International Conference on* (pp. 47-52). IEEE.

Wu, Z., Meng, Z., & Gray, J. (2017). MANET-based Techniques for Online M2M: Interactive itemised data registration and offline information traceability in a digital manufacturing system. *IEEE Transactions on Industrial Info*dustrial Informatics.

T. Savolainen, J. Soininen, B. Silverajan, "IPv6 addressing strategies for IoT", *IEEE Sensors J.*, vol. 13, no. 10, pp. 3511-3519, Oct. 2013.

Munns, C., & Basu, S. (2015). *Privacy and healthcare data: 'Choice of Control' to 'Choice' and 'Control'*. Farnham: Ashgate Publishing.

Mehta, R., Kale, S., & Utage, A. S. (2017). The internet of Things (IoT) intelligence computing technology for home automation. *International Journal of Current Engineering and Technology*.

Breur, T. J. (2015). Big data and the internet things. *Journal of marketing an analytic, 3*(1), 1-4.

Kumar, A. D., & Venugopala, S. R. (2017). Intrusion detection by initial classification-based on protocol type *International Journal of Advanced Intelligence Paradigms 9*(2/3), 122.

Kong, H., Jong, C., & Ryang, U. (2016). *Rare Association Rule Mining for Network Intrusion Detection.*

Sahasrabuddhe, A., Naikade, S., Ramaswamy, A., Sadliwala, B., & Futane, P. (2017). Survey on Intrusion Detection System using Data Mining Techniques. *International Research Journal of Engineering and Technology 4*(5), 1780.

.