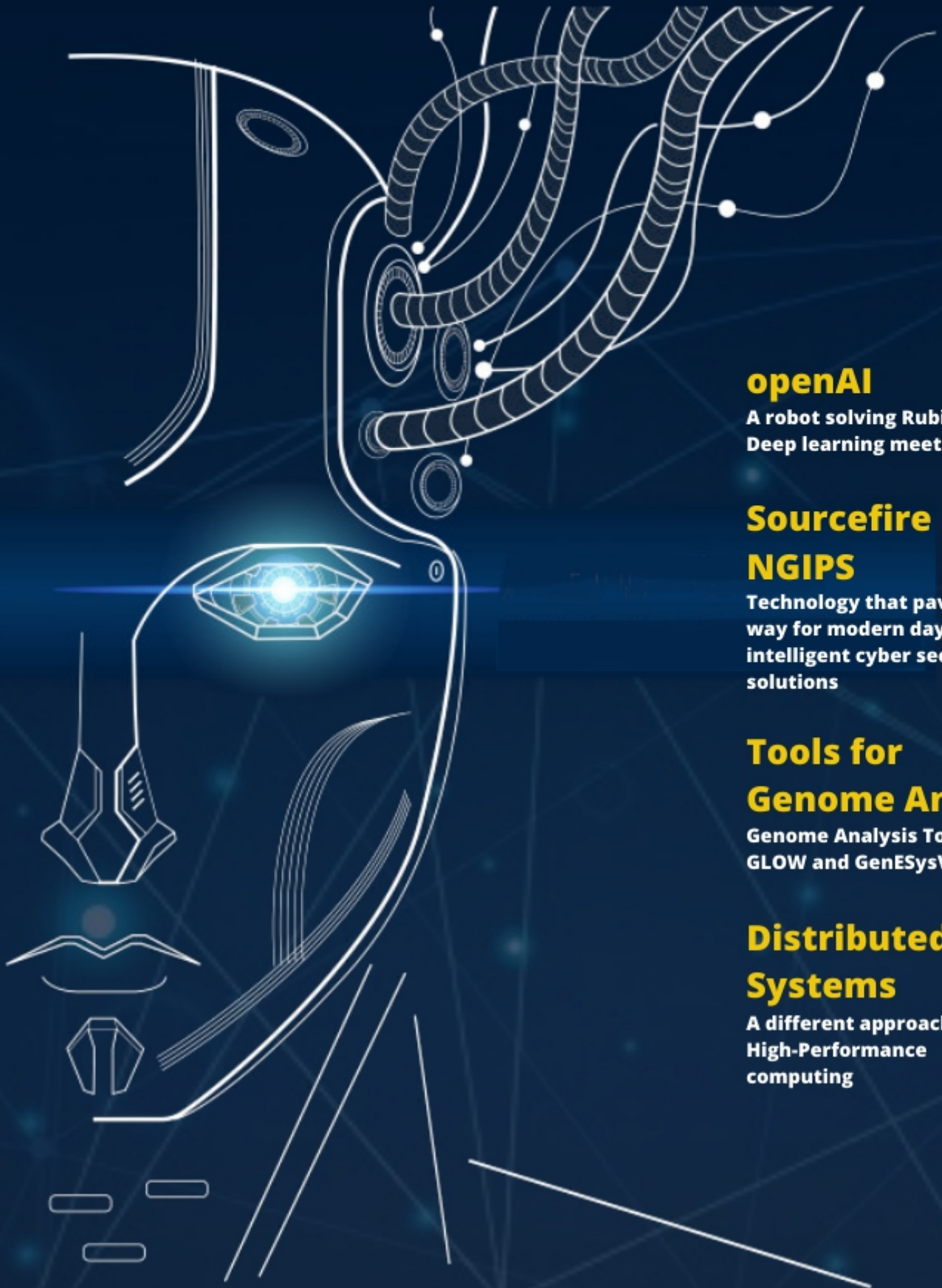
IEEE INTERFACE

Volume 18
Issue 53



openAI

A robot solving Rubik's Cube : Deep learning meets Robotics

Sourcefire NGIPS

Technology that paved the way for modern day intelligent cyber security solutions

Tools for Genome Analysis

Genome Analysis Toolkit, GLOW and GenESysV

Distributed Systems

A different approach to High-Performance computing

CONTENTS

Hello!

The Editorial Board of IEEE-SJCE is back with the 53rd issue of INTERFACE for Tuxedo, 2019. With the Tuxedo edition of this magazine, we always aim to inspire more of our readers to venture into Open Source Technologies. By flipping through the pages, you can read about some cutting edge open source softwares, AI with Robotics, Optronic sensors among many others. In addition to this, you also get some insights about the events conducted by IEEE-SJCE's student branch so far.

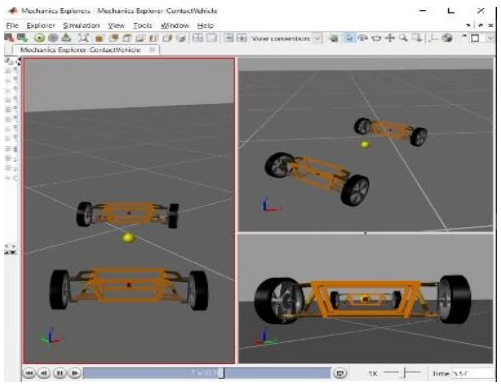
Cheers!

-The IEEE-SJCE Ed Board

Deep learning meets robotics - the openAI way!

-Akshay Krishnan, IEEE-SJCE alumnus

Robotics is usually seen as a confluence of Artificial Intelligence, perception and control. One acceptable definition of a robot is a machine that can intelligently make decisions (Artificial Intelligence), by reasoning about data from the surroundings (Perception) to perform mechanical tasks - say manipulating a Rubik's cube (Control theory). Conventional control or perception algorithms assume a precise mathematical model of the environment. In case of a robotic arm, this could be a mathematical model that relates the voltages applied on the actuators to the velocity of a finger. The problem with this - as harsh as it may sound, is that not everything in the real world fits into a mathematical model.



Enter machine learning! Deep learning (a subset of machine learning) provides us with a universal function approximator. This essentially means that any function can be described by a graph that can be "learnt" using data (inputs and outputs of the function that is being approximated). This way, if we can collect enough data we can learn a graphical model (also popularly known as a neural network or a deep network), to approximate the control algorithm and the perception algorithm. These learnt deep networks have proven to perform better in the real world compared to classical algorithms for many robotic tasks. However, there are many challenges associated with training deep networks - one of them being the requirement of labelled data (input data annotated with the required output). One way to get around this requirement of labelled data is train these deep models in a simulator instead of training on the real robot. Labelled data is available in plenty in when using a simulator (I'll leave you to guess why).

This is what OpenAI exploited in their recent project, where they got a robot to solve a Rubik's cube. But before we get into that, let's talk a little about OpenAI. OpenAI is an organization based in San Francisco that was founded by a team of AI enthusiasts (one of them being Elon Musk) in order to push the limits of AI with ground-breaking research that is eventually open-sourced. To date, they have made significant contributions to the AI research community mainly in the area of Reinforcement learning - including my personal favorite - the OpenAI Gym simulator for training reinforcement learning agents.

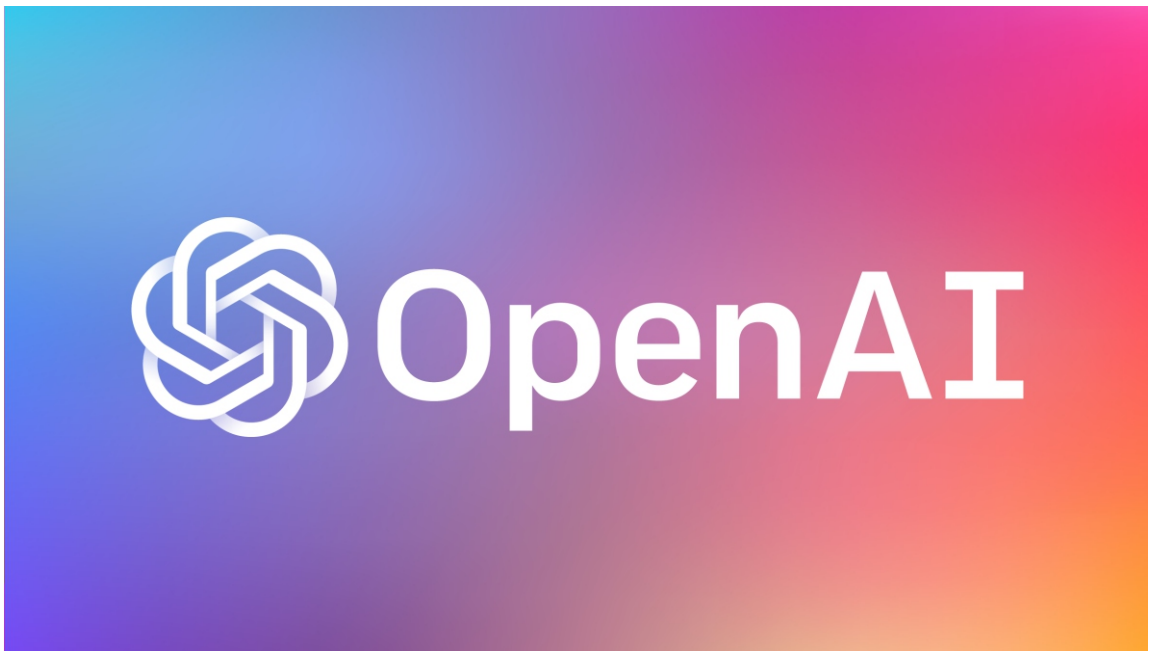
In their recent project, OpenAI used deep reinforcement learning to train a model (controller) for the robot arm that can solve the Rubik's cube. There are two aspects of this problem that needs to be solved - estimating the current state of the cube using inputs from cameras, and estimating the control inputs to the arm in order to move a particular "face" of the cube. Notice that I did not mention deciding which face is to be resolved as a problem first. This is because this problem has long been solved and there exist software that can do this easily. For the two other problems, openAI uses deep learning to train a neural network that can estimate the pose of the cube in the robot's hand, and another neural network to represent a reinforcement learning policy that controls the action of the robot arm when the state of the cube and the face to be moved is known.

The intuition behind domain randomization. The neural network is trained on varying settings in the simulator before it is used in the real world!

All of this together helped OpenAI build a robot arm that can solve a Rubik's cube. I would challenge you to do it yourself, just to appreciate the difficulty of the task! This is just one of the many success stories that deep reinforcement learning has seen in the area of robotics. Companies like Google Brain, NVIDIA, and Facebook Research have also been very active in this research area. Will this trend of learning-based methods replacing classical model-based approaches pave the way for truly intelligent machines in the future? Time alone can tell!

Every component mentioned above has been addressed by previous research. The novelty in this project is the way OpenAI generates data. I mentioned earlier that one of the challenges of deep learning is the shortage of labelled data, and this can be alleviated by using data generated in simulation. A problem with doing so is that there is no guarantee that a neural network that is trained on data generated on simulation will perform well in the real world. Theory states that a machine learning model can perform well, if the unseen data is from the same probability distribution that generated the training data. However, this is no longer true when migrating from simulation to the real world.

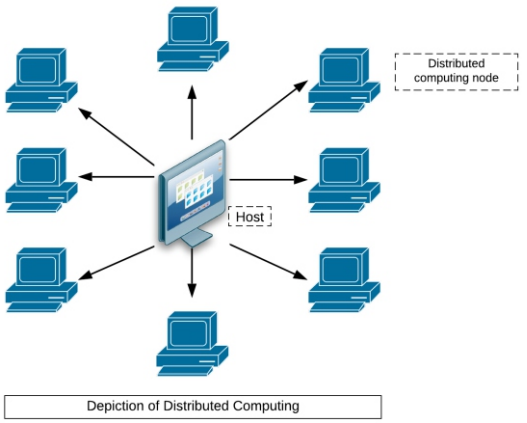
So OpenAI uses a technique called "Domain randomization". Consider a scenario where you are learning to play cricket using a tennis ball but are suddenly asked to play a cork ball match (I know, we all have been there). You are bound to get hurt in your first game. However, if you had learnt to play cricket using different kinds of balls (but not the cork ball), including say one made of stone, you would have better chances of performing well in your first cork ball game. That is of stone, you would have better chances of performing well in your first cork ball game. That is the intuition behind domain randomization. The neural network is trained on varying settings in the simulator before it is used in the real world!



A dive into Distributed Systems

-Naresh Prabhu, MS, GeorgiaTech

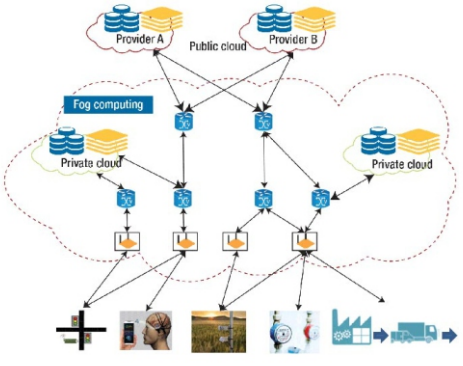
In my previous article titled General Purpose Programming on GPU, I tried to make a case on how GPUs are going to be the High-Performance Computational devices in the days to come – those days are here already – and deliver significantly more performance capabilities and value to the user. This article is going to be about High-Performance Computing again, with a slightly different approach – Distributed Computing.



Depiction of Distributed Computing

In the conventional way of computing, what we're accustomed to is - using a machine – computer – we provide some input, it processes the input, and spits out some output. This in fact is a very standard interaction that most of us experience in our everyday lives. And it turns out that this kind of approach will suffice to get done most of our regular, day to day tasks that need a computer (by computer, I mean smartphones too) – sending & receiving emails, stalking people on Instagram, watching allowed & unallowed content on the internet...you get it. But when we're trying to do a processor intensive project like 3D graphics rendering, simulating a weather model or trajectory of a hurricane, or, a more recently achieved remarkable feat like building the image of a black hole from the data collected, the chances are that in such cases the processing power of a single computer may not be enough – single computer might be painfully slow to solve such a computationally intensive operation – and that is where our protagonist Distributed Computing comes in. Actually, the idea of distributed computing is really simple (only the idea is simple) – you take a large, complex task, chop it up into smaller tasks, and distribute the workload over a large number of computers so that each computer only needs to munch through small tasks. As all the computers are working together, the consequence of this action is the result of your large computation task (in a far less time).

The real difficulty comes when a task is to be chosen for computing by distribution. Certain tasks are not very suitable for the purposes of distributed computing – they can be like some task that cannot be divided into smaller chunks, or, a large task that needs to be computed in a single large step – and they cannot be distributed as smaller tasks to a large number of computers. Another example of a difficult task for the sake of distributed computing is a task with many steps, and every successive step depends on the result of the previous step. Once a serial pattern like this gets involved, it becomes much harder to solve this problem using distributed computing. So, by now if you're thinking that, if tasks with a heavy serial component may not be suitable for distributed computing, does that mean tasks which can be parallelized, are perfect for distributed computing then, you're absolutely right. Parallelizable tasks are those tasks which are independent of each other and can be executed simultaneously.



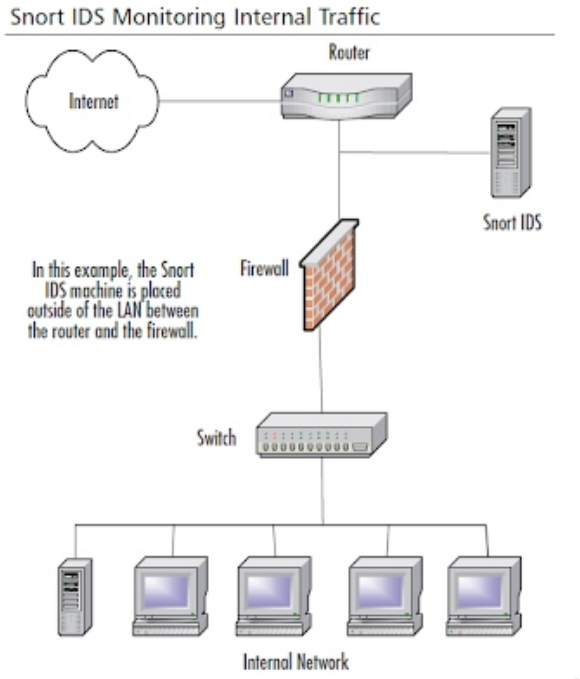
The way distributed computing is done is simple (well, not really). Basically, you have a host computer, as well as an ensemble of computers that are going to help us with distributed computing. The host computer is the computer with the help of which you will control the functionalities of other computers. In host computer, you setup your task by running your primary program which is responsible to divide your large computational piece into smaller jobs, distribute these jobs to the rest of the computer, and once the computation is done by each computer, combine the results of all the jobs to generate the final result. As you may have guessed, there is a communication going on between the host computer and the rest of the computers, and also, there might be some communication going on among the rest of the computers themselves. Now, this happens over a network – a network that might be similar to the one in our homes like LAN, or a network similar to that of the enterprises like WAN.

So, at this point we have a macro-level understanding of what distributed computing is. All I can say about this field of computing is that, as more and more computationally intensive tasks like training a self-driving car algorithm, economy and stock forecasting, blockchain, cryptography, space exploration start becoming ubiquitous, the need for engineers who can take the responsibility of working in this domain is going to constantly increase. Though distributed computing exists and is being used since more than two decades, there is a dearth in human resource for this field (one reason is that it is expensive to access all the resources required to learn it). Thus, if you're looking for name, fame, money, and want to stay away from huge crowd that is migrating towards the current day buzz, that is, artificial intelligence, machine learning, deep learning, and data science, you can choose to take the path of a distributed computing engineer and play a key role in building technology that is an enabler of each one of these technologies.

Sourcefire NGIPS

-Shreya A N, VII Sem, ISE

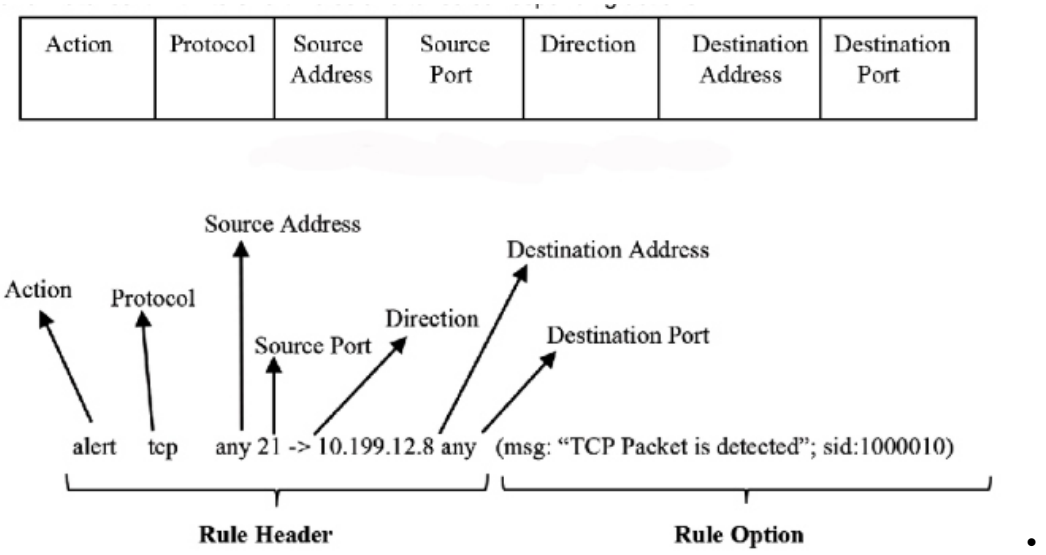
With the rapid proliferation of networks, most of the world today is connected. While this has proved to be of significant importance, it has also empowered hackers with malicious intents. An organisation's critical systems housing confidential data are often targeted. Hackers attack networks either actively or passively, with the intention of causing harm directly or leaking classified information to the public. The need to safeguard our networks now is more than ever. This is where Intrusion Detection Systems(IDS) and Intrusion Prevention Systems(IPS) come into play.



In common terms, an IDS is a software that monitors traffic flowing over a network for suspicious packets, policy breaching and alerts the administration when such activities are detected. An IPS is an upgrade from IDS as it can not only detect irregularities but can also take rational decisions to prevent them. In network topology, IDS and IPS usually sit behind a firewall and analyse packets before forwarding them to their destinations(in case of no anomalies). For this reason, IDS and IPS are grouped under inline network devices, alongside firewalls, routers and switches.

In 1998, Martin Roesch developed an open-source, lightweight IDS and named it Snort. Snort is a packet sniffer and closely examines each and every packet passing through the IDS. Through extensive protocol analysis, Snort can successfully detect Denial of Service(DoS) attacks, bottlenecks and buffer overflow. It works in association with a database containing details about what traffic needs to be dropped. This information collectively forms Snort Rules. Snort rules contain 7 fields viz. source IP address, source port address, destination IP address, destination port, protocol, action and direction of flow. Snort collects information from the incoming packets and matches it with its Snort Rules and takes corresponding actions.

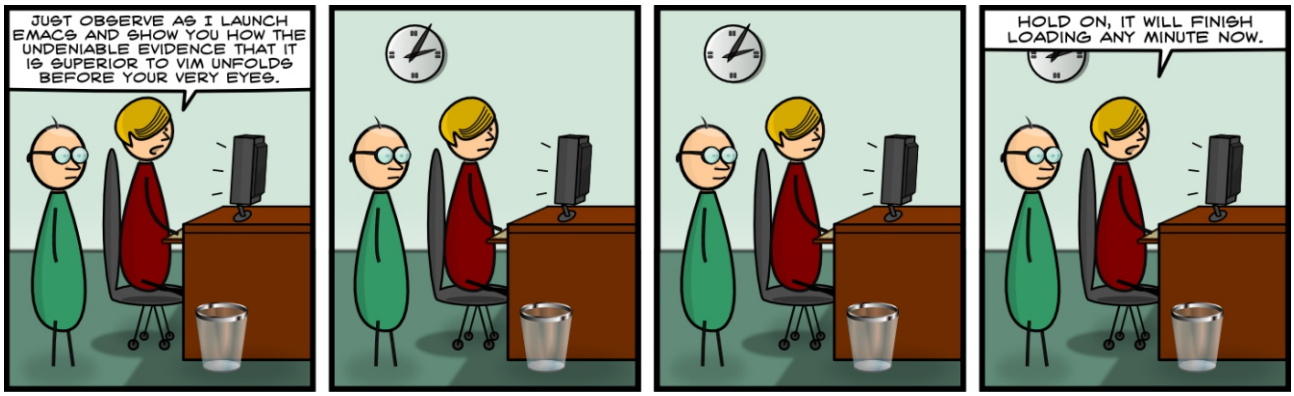
A European network security testing organization NSS, Network Security Services, tested Snort along with intrusion detection system (IDS) products from 15 major vendors including Cisco, Computer Associates, and Symantec, and declared



Snort as the sole open-source freeware product tested that clearly out-performed other proprietary products. In 2001, Roesch founded his own company named Sourcefire that developed network security hardware and software. The company released a new line of intelligent security appliances called Sourcefire Firepower by combining Snort and IDS technologies. In 2013, Cisco, the American multinational IT and Networking technology conglomerate, acquired Sourcefire for $2.7 billion. Since then, Snort is being developed and maintained by the developer community at Cisco.

Today, Snort, with over 5 million downloads and over 600,000 registered users, is the most widely deployed intrusion prevention system in the world. Needless to say, Martin Roesch's Sourcefire remained the leader in intelligent cybersecurity solutions for more than a decade and paved the way for modern-day Next Generation Intrusion Prevention Systems(NGIPS).

Sources: Cisco Acquisitions, Wikipedia, Harrykar's Techies Blog, Semantic Scholartech.



A 411 to Solid State Drives (SSDs)

-Aneesh Kalkur, 5th Sem, BT

Figure showing a Samsung SSD, one of the most popular SSDs in the market today.

I'm sure everyone has had to buy a laptop or a desktop at some point in time. In order to purchase one, we'd have seen the features of the desktop or the laptop computer. Some of these features include the display type, its size, and processor speed, RAM, storage capacity, battery life and many more to come.

We also happen to come across a new form of secondary storage memory, besides our humble Hard Disk Drive (HDD) whose use tends to make our computers boot up much faster than a computer which only has a hard disk drive. In fact, some devices like our Apple Macintosh have made HDDs a thing of the past and replaced it with solid state drives, more popular as SSDs. Now with enough introduction, let us know all that we can and need to know about SSDs. SSDs are solid-state storage devices that use integrated circuit assemblies as memory, in order to Store data persistently using flash memory. The concept of SSDs was first proposed by Storage Technology Corporation (STC), in 1978 and it had a memory of 45MB, which was high for the time. With developments in flash memorytechnology in the 1980s, SunDisk (now SanDisk) developed the first commercial SSD and started shipping it in 1991 with a size of 20MB.

SSDs, just like RAM, use semiconductors to store memory. But unlike HDDs, SSDs do not rely on all the mechanical platters and disks in order to store long-term memory. Instead, we have multiple transistors in a grid, called cells throughout it. Based on this, we have two types of flash memory used in SSDs: NOR, where the cells are wired in parallel, and NAND, where the cells are wired together in series. The latter is ideal for commercial use for it has a smaller form factor, and can handle a higher density of data. This feature also makes it relatively inexpensive compared to NOR-based SSDs. SSDs use logical gates to store memory as 1s and 0s. If the current passes through the cell, it has a value of 1, else the value in the cell becomes 0. At each intersection of the column and row, two transistors form a cell. One of the transistors is known as a control gate, the other as a floating gate. When current reaches the control gate, electrons flow onto the floating gate, creating a net positive charge that interrupts current flow. On weighing the pros and cons of SSDs, we conclude that both SSDs and HDDs have a place to stay, at least for now. It would be recommended to use SSDs as our boot drive and HDDs as our long-term storage drive. every desktop or laptop comes fitted with solid-state drives, making HDDs, a thing of the past.

By applying precise voltages to the transistors, a unique pattern of 1s and 0s emerges, allowing memory storage. SSDs come with several advantages over HDDs. Its use as a boot drive in the computer dramatically reduces the time taken by the system to load applications present in the system. This upgrade can massively increase transfer speeds, even with existing computers. In fact, SSDs are made to look similar to HDDs so they can fit into their slots. SSDs can also be used as a replacement for our existing HDDs: as mentioned earlier, some laptops only come with SSD storage. Also, SSDs lack moving parts, and so, can operate in absolute silence. However, SSDs have longevity issues that HDDs do not get affected with, even after years of use. This happens because they cannot write a single bit of information without erasing and rewriting very large blocks of data at once. Each time the cell is erased of data, the floating gate transistor is left with charge, causing a change in its resistance. This builds up, forcing the need for more current to pass through the cell.

Eventually, the cell becomes so resistant to current that it cannot be flipped, thereby prematurely ending the life of the cell. However, SSDs come with wear-levelling abilities, so the wear across the cells present on the SSD are evenly distributed. This can be used to provide longevity for the cell. Besides that, it needs to be used as often as possible for years of storage can cause leakage of charge, thereby erasing the data present in this drive. On weighing the pros and cons of SSDs, we conclude that both SSDs and HDDs have a place to stay, at least for now. It would be recommended to use SSDs as our boot drive and HDDs as our long-term storage drive. However, we have come to a point where SSDs are more easily available at affordable prices, and are more durable than they used to be, back when they first came out in the 1990s. With improvements in solid-state flash memory occurring every day, we hope to see a day where every desktop or laptop comes fitted with solid-state drives, making HDDs, a thing of the past.

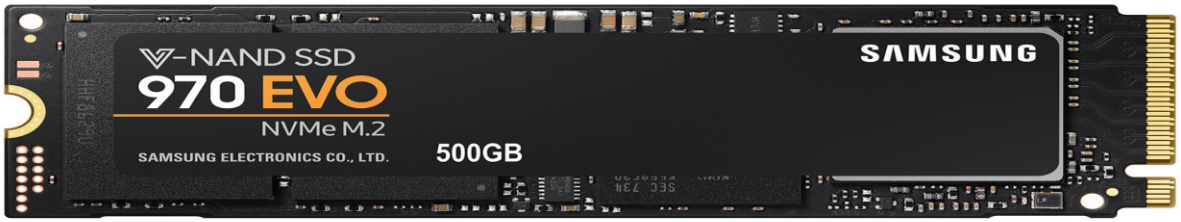


Fig: 970 EVO SSD - SAMSUNG

PageRank Algorithm

-Susmitha Raj, 7th Sem, ISE

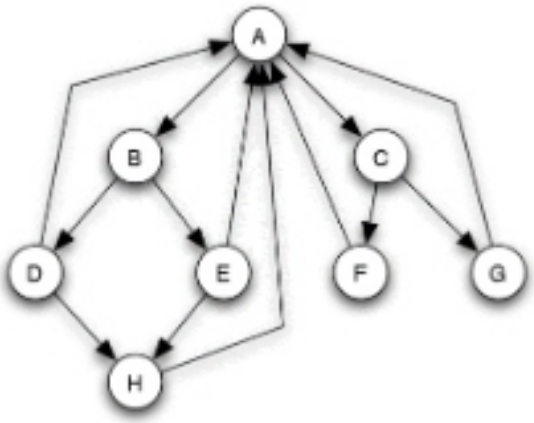
PageRank is an algorithm used to rank web pages in the search engine results. It works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that important websites are likely to receive more links from other websites. Consider someone who is randomly browsing a network of Web pages, he starts by choosing a page at random, picking each page with equal probability. He then follows links for a sequence of k steps. In each step, he picks a random out-going link from their current page, and follows it to where it leads. (If their current page has no out-going links, he just stays where he is.) Such an exploration of nodes(pages) performed by randomly following links is called a random walk on the network. Considering this random walk on the network, PageRank is computed as follows.

1. In a network with n nodes, all nodes are assigned with the same initial PageRank, set to be 1/n.

2. Number of steps k is chosen

3. Sequence of k updates are performed to the PageRank values, using the following rules for each update:

RULE 1: Basic PageRank Update Rule: Each page divides its current PageRank equally across its out-going links, and passes these equal shares to the pages it points to. (If a page has no out-going links, it passes all its current PageRank to itself.) Each page updates its new PageRank to be the sum of the shares it receives. Total PageRank in the network remains constant.

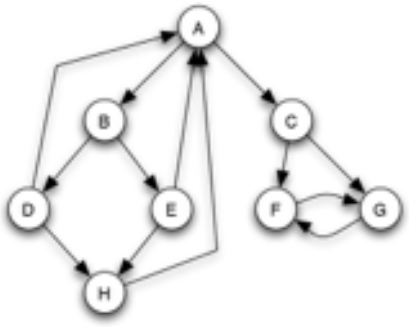


Since there are 8 nodes in the graph, all are assigned the initial PageRank value of 1/8. Their PageRank values after the first two updates are given by the following table:

Step	A	B	C	D	E	F	G	H
1	1/2	1/16	1/16	1/16	1/16	1/16	1/16	1/8
2	3/16	1/4	1/4	1/32	1/32	1/32	1/32	1/16

For example, A gets a PageRank of 1/2 after the first update because it gets all of F's, G's, and H's PageRank, and half each of D's and E's. On the other hand, B and C each get half of A's PageRank, so they only get 1/16 each in the first step. But once A acquires a lot of PageRank, B and C benefit in the next step. This is in keeping with the principle of repeated improvement: after the first update causes us to estimate that A is an important page, we weight its contribution more highly in the next update. After a specific number of updates the PageRank values get converged.

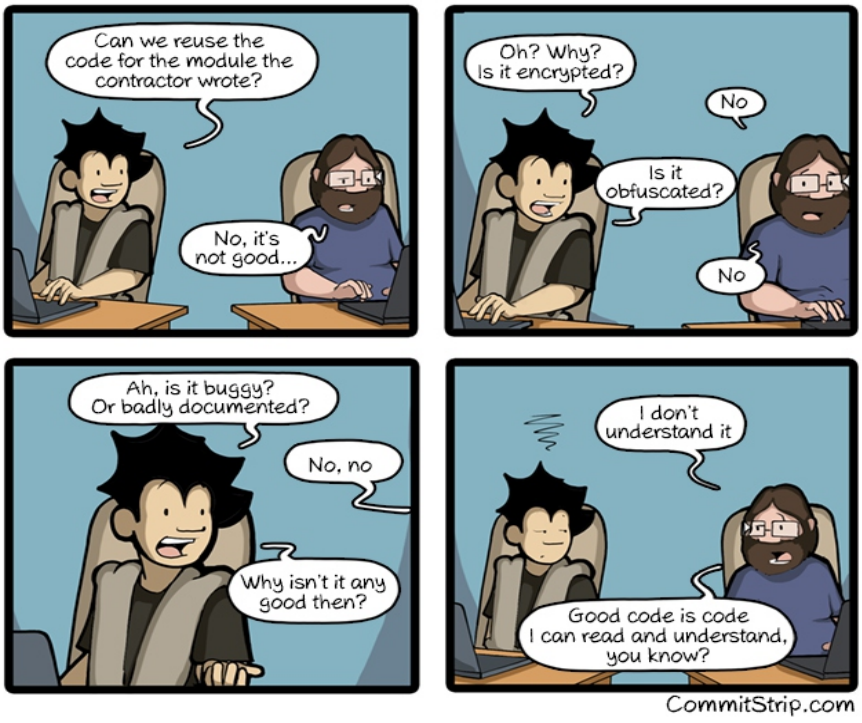
Let's make a small change in the figure, so that F and G now point to each other rather than pointing to A.



Clearly, this thought to weaken A somewhat, but in fact a much more extreme thing happens: PageRank that flows from C to F and G can never circulate back into the rest of the network, and so the links out of C function as a kind of "slow leak" that eventually causes all the PageRank to end up at F and G. We can indeed check that by repeatedly running the Basic PageRank Update Rule, we converge to PageRank values of 1/2 for each of F and G, and 0 for all other nodes.

This is clearly not what we want. In order to avoid the above mentioned problem, we pick a scaling factor 's' that should be strictly between 0 and 1. We then use Scaled PageRank Update Rule along with Basic PageRank Update Rule.

RULE 2: Scaled PageRank Update Rule: First apply the Basic PageRank Update Rule. Then scale down all PageRank values by a factor of s. This means that the total PageRank in the network has shrunk from 1 to s. We divide the residual 1−s units of PageRank equally over all nodes, giving (1−s)/n to each. This rule eliminates the accumulation of PageRank at only few specific nodes. Repeated application of the Scaled PageRank Update Rule converges to a set of limiting PageRank values as the number of updates k goes to infinity. Moreover, for any network, these limiting values form the unique equilibrium for the Scaled PageRank Update Rule: they are the unique set of values that remain unchanged under the application of this update rule. These values depend on our choice of the scaling factor s which is usually chosen to be between 0.8 and 0.9.



VisibleV8

-Devraj.R, 5th sem, CSE

Researchers have developed an open-source tool that allows users to track and record the behaviour of JavaScript programs without alerting the websites that run those programs. The tool, called VisibleV8, runs on the Chrome browser and is designed to detect malicious programs that are capable of evading existing malware detection systems.

“When you visit most websites, your browser starts running the site's JavaScript programs pretty much immediately -- and you have little or no idea of what JavaScript is doing," says Alexandros Kapravelos, co-author of a paper on VisibleV8 and an assistant professor of computer science at NC State. "Previous state-of-the-art malware detection systems rely on making changes to JavaScript code in order to see how the code is being executed. But this approach is easily detected, allowing malware programs to alter their behaviour in order to avoid being identified as malicious”.

“VisibleV8 runs in the browser itself, recording how JavaScript is executed; it doesn't interact with the code and, as a result, is far more difficult to detect”.

VisibleV8 saves all of the data on how a site is using JavaScript, creating a "behaviour profile" for the site. That profile and all of the supporting data can then be used by researchers to identify both malicious websites and the various ways that JavaScript is used to compromise web browsers and user information.

Because VisibleV8 consists of only 600 lines of code, out of the millions of lines of code in Chrome, the software tool is relatively easy to keep up-to-date. This is an important consideration given that Chrome's code is updated approximately every six weeks. VisibleV8 can also be used to target the most likely malicious behaviours without hurting browser performance.

“We've created a stealthy tool for monitoring JavaScript in the wild," Kapravelos says. "We're now making it open source, in hopes that it will be useful to anyone doing research on web privacy and security."

Genome Analysis Tools

-Aditi Jayakumar, 7th Sem, BT

Millions of chemical bases make up the genome and are encoded using four bases- A (Adenine), T (Thymine), G (Guanine), C (Cytosine). The arrangement of these bases is important as they decide the functionality of gene and therefore the concept of genome analysis was introduced.

Genome analysis is the key solution to most of the medical challenges faced in the current world. Understanding the impact of minor changes or mutation on the health of an individual requires the sequencing of the desired gene. Once sequenced, the data can be inspected to create models that can predict risk factors and also personalize the medicine, targeting a single defective gene to attain maximum possible therapeutic effect for each individual. Given the size of the human genome, decoding the gene organization, its functions, interactions, remains a Herculean task. Traditional methods of sequence analysis require months of work and yet can not be completely relied upon. Hence bioinformatic tools are being used for accurate scientific results.

Next-generation sequencing (NGS) is one such breakthrough concept that has led to the understanding of genetic variation among individuals very effectively. NGS uses the concept of massively parallel processing. It's a powerful tool to sequence millions of DNA simultaneously saving an ample amount of time. These technologies use parallelized platforms for sequencing of 1 million to 43 billion short reads (50-400 bases each) per instrument run. However, the massive data sets generated by NGS require analysis tools for accessing and manipulating the data produced.

Genome Analysis Toolkit (GATK) is a structured programming framework designed to ease the development of efficient analysis tools for next-generation DNA sequencers using the functional programming. It provides an option for the developers to create analysis specific algorithm which helps in processing complex NGS data. The GATKS's efficiency has allowed the tools to be deployed for various international projects such as HLA typing, multiple-sample SNP (Single Nucleotide Polymorphism) genotyping, indel discovery, etc. In 2017, the Broad's Institute decided to make the software package an open source to allow the community to move ahead efficiently and collaboratively.

Glow is also an open-source software toolkit built on Apache Spark (cluster-computing framework). Some genes can be mapped by the physical or visual changes associated with its function. These genes when mutated lead significant morphological or physical differences called the phenotypic variation. GLOW provides an excellent method to aggregate the genotypic and phenotypic data. It uses CNN and Machine Learning to train models to identify SNPs, mutation along with BioImaging techniques. The algorithms to simplify genome data preparation, and statistical analysis at bio-bank scales. It has built-in, single-line commands in Python, R, Scala, and SQL for common genomic data interpretation and analysis.

GenESysV was the first tool to be able to handle variant datasets ranging in size from a few to thousands of samples and maintain fast data importation. It is user-friendly and scalable. Many variants underlying Mendelian and complex diseases are being discovered and documented using these techniques. It is designed for the use of researchers analyzing high throughput sequencing for efficient retrieval of variants of interest.

The above listed are few open-source toolkits out of many existing powerful softwares. These tools can be used to manipulate the data to give unimaginable results. Such technologies help the research community to speed up the discoveries for mechanisms underlying genetic disorders.

LINUX

-Sahana Gowda, 7th Sem, CSE

Linux is a family of free and open-source software operating systems built around the Linux kernel. It is a very popular open-source operating system that runs on all major hardware platforms including x86, Itanium, PowerPC, ARM and IBM mainframes.

Tux is a penguin character and the official brand character of the Linux kernel. Originally created as an entry to a Linux logo competition, Tux is the most commonly used icon for Linux. Tux was originally drawn by Larry Ewing in 1996.

Linux, like Mac OS X, is based on the Unix operating system. The Unix operating system was conceived and implemented in 1969, at AT&T's Bell Laboratories in the United States by Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe Ossanna. First released in 1971, Unix was written entirely in assembly language. Later, in a key pioneering approach in 1973, it was rewritten in the C programming language by Dennis Ritchie. The availability of a high-level language implementation of Unix made its porting to different computer platforms easier.

Strictly speaking, Linux is the kernel, not the entire operating system. The kernel provides an interface between your Linode's hardware and the input/output requests from applications. The rest of the operating system usually includes many GNU libraries, utilities, and other software, from the Free Software Foundation. The operating system as a whole is known as GNU/Linux.

Linus Torvalds is the creator of Linux who released the Linux kernel as free, open-source software (Open source means that the code is fully visible, and can be modified and redistributed). In 1991, while attending the University of Helsinki, Torvalds became curious about operating systems. Frustrated by the licensing of MINIX, which at the time-limited it to educational use only, he began to work on his own operating system kernel, which eventually became the Linux kernel. He had wanted to call his invention "Freax", a portmanteau of "free", "freak", and "x" but later called it Linux.

At present Greg Kroah-Hartman is the lead maintainer for the Linux kernel and guides its development.

The primary difference between Linux and many other popular contemporary operating systems is that the Linux kernel and other components are free and open-source software. Linux is not the only such operating system, although it is by far the most widely used. The most common free software license, the GNU General Public License (GPL), is used for the Linux kernel and many of the components from the GNU Project.

Linux-based distributions are intended by developers for interoperability with other operating systems and established computing standards. Linux systems adhere to POSIX, SUS, LSB, ISO, and ANSI standards where possible, although to date only one Linux distribution has been POSIX.1 certified, Linux-FT. Many Linux distributions, or "distros", manage a remote collection of system software and application software packages available for download and installation through a network connection.

Besides the Linux distributions designed for general-purpose use on desktops and servers, distributions may be specialized for different purposes including computer architecture support, embedded systems, stability, security, localization to a specific region or language, targeting of specific user groups, support for real-time applications, or commitment to a given desktop environment.

Optronic Sensors

-Harsha Bhat, 3rd sem, EIE

Fundamentals and types:

Understanding optronic sensors and related sensor systems is essential in understanding the design and operation of a large number of defence systems.

Optronic sensors constitute the heart of a variety of systems ranging from simple gadgets like light meters to the most complex of military systems like precision-guided munitions, laser rangefinders, target trackers, remote sensing systems, navigation sensors, sniper and explosive detectors, fibre-optic and laser-based communication systems, night vision devices, Lidar and many more.

While individual photosensors such as PIN photodiodes and avalanche photodiodes find applications in military optronic systems including laser rangefinders and target designators, Lidar sensors and Navigation sensors, sensor array such as Complementary Metal Oxide Semiconductor (CMOS), charge-coupled device and avalanche photodiodes are at the core of imaging sensor systems including Ladar sensors, night vision devices and imaging infrared seekers.

Types of Photosensors:

Photosensors are classified into two major categories: Photoelectric and Thermal.

Photoelectric sensors are further of two types: Devices that depend on external photo effect for their operation and devices that make use of some kind of internal photo effect.

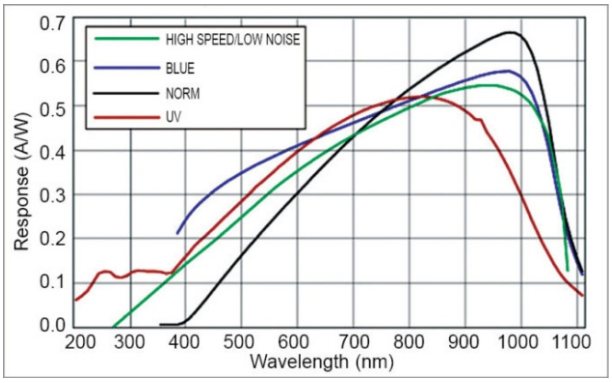
Photo-emissive sensors are based on external photo effect. Common photo-emissive sensors include non-imaging sensors such as vacuum photocells and photomultiplier tubes, and imaging sensors such as image intensifier tubes.

Major parameters used to characterise the performance of photosensors include responsivity, Noise Equivalent Power, Sensitivity, Quantum efficiency, Response time and noise.

Responsivity is the ratio of electrical output to radiant light input determined in the linear region of response. It is measured in A/W.

Silicon photodiodes exhibit a response from ultraviolet through invisible and into near-infrared part of the spectrum.

With silicon having a band gap energy of 1.12eV at room temperature, its spectral response peaks in the near-infrared region between 800nm and 950nm.



Photoconductors:

Photoconductors are also referred to as photoresistors, light-dependent resistors(LDRs) and photocells-- are semiconductor photosensors whose resistance decreases with increase in incident light intensity. These are bulk semiconductor devices with no pn junctions.

Commonly used materials for photoconductors are cadmium-sulphide(CdS), lead-sulphide (PbS), etc.

Some representative packages of semiconductors:

Photodiodes:

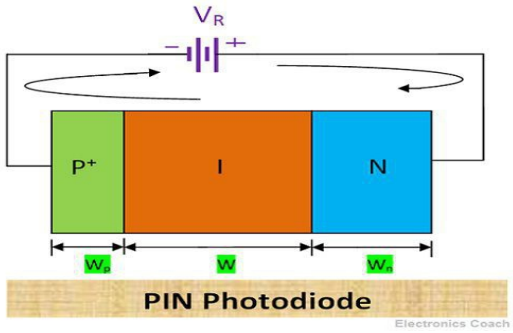
Photodiodes are junction-type semiconductor light sensors that generate current or voltage when p-n junction in the semiconductor is illuminated by light of sufficient energy.

Types of photodiodes:

Photodiodes and Avalanche photodiodes.

p-n photodiodes comprise a p-n junction. When light with sufficient energy strikes, its electrons are pulled up into conduction band, leaving behind holes in the valence band.

These electron holes pair occur throughout p-layer, depletion layer and n-layer. When the photodiode is reverse biased, photo-induced electrons move down the potential hill from p to n side. Similarly, photo-induced holes add to the current flow by moving across the junction to the p-side.



In PIN photodiodes, an extra high resistance intrinsic layer is added between p and n layers. This reduces the transit or diffusion time of photo-induced electron-hole pairs, which in turn, improves the response time.

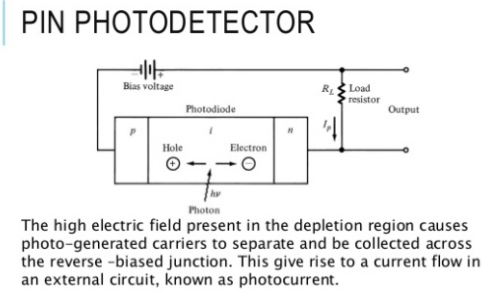
PIN photodiodes feature low capacitance and therefore high bandwidth, which makes them suitable for high-speed photometry as well as optical communication applications.

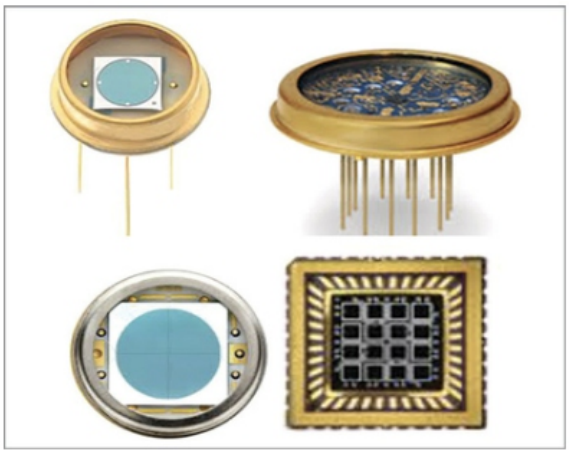
While the charge separation also occurs without the application of an external voltage, the process can be accelerated by such a reverse voltage. The photocurrent remains linear to the absorbed light volume across many orders of magnitude if the diode is not operated in a state of saturation.

Depending on the external connections, we differentiate between two different operating states: element and diode. In the case of element operation, the diode is connected directly to the consumer without the use of an external voltage source. No dark current flows in this operating state, which facilitates the detection of minimal intensities. During diode operation, an external voltage supply is connected with the consumer in series, whereby the voltage is applied in reverse direction. This operating mode is ideal for applications in which a rapid signal response is required. The main disadvantage is the dark current, which grows exponentially with the temperature.

A PIN diode comprises a near-intrinsic semiconductor region – usually the space-charge region – sandwiched between a p-type diode and an n-type substrate. However, the term is also used for components with inverse conductivity, provided that no other non-linear effects are utilized in the component.

Fig: PIN PHOTODECTOR





Meanwhile at IEEE...

IEEE is always actively conducting workshops to help students grow as professionals and engineers. Many such events are conducted every year. Here are a few highlights from the last semester-

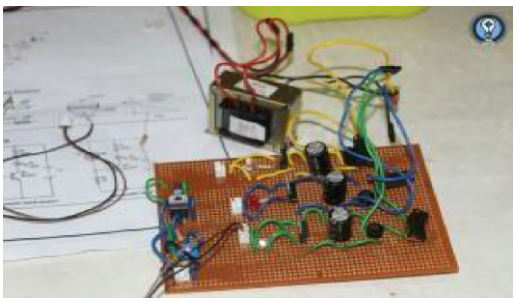
VPM 10.0

Vacation Project Mania (VPM) is an inclusive training program which makes students think outside the box, develop in-depth practical knowledge about the subject and gain skills which come in handy while executing challenging projects. The workshop encourages students to engage in real-time projects and endorses teamwork. The participants formed their own team to execute the various tasks assigned to them during the workshop. The students were encouraged to provide constructive and innovative solutions to overcome several technical hitches resulting in a good project.

IEEE-SJCE has successfully conducted VPM for 9 years. Several students have skipped their vacation to take part in the 10th edition of VPM which was held from 12th to 19th June 2019. The program was divided into three phases.

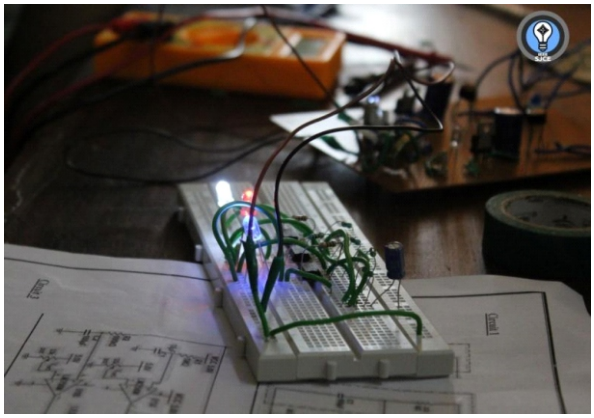
Phase 1: Regulated Power Supply and Variable Power Supply

This phase was aimed at providing a deeper insight into basic analog electronics and simple circuits that are used in a majority of applications. One of the major goals of this session was to improvise the technological know-how. The concepts were revised and discussed with participants in the theory sessions which were followed by the practical sessions which comprised of several hands-on activities. In addition to this, the participants were introduced to designing circuits as it plays a major role in the effective working of any device.



Phase 2: IR pair, Opamps, LDR, Diodes

The major area of focus in phase 2 of the workshop was basic concepts of IR pairs, LDRs, diodes and Opamps. Participants were able to build basic circuits in the practical sessions.



Phase3: Hackathon and Project Submissions

On the last day of the workshop, the participants actively took part in the Hackathon where they were given challenging tasks that were to be solved based on the knowledge they gained during the course of the workshop. This competition aimed to encourage team activity and collective effort which plays a significant part in a project. The teams came up with innovative and smart solutions to the tasks given, thereby leading to close competition. Two teams bagged prices for completing the tasks effectively.

This phase also involves Project submission. The major takeaways of the workshop are strengthening of basics, enhanced problem-solving abilities, better practical knowledge. VPM thus provides a platform for participants to showcase these qualities by submitting projects after a period of one month. In order to encourage the juniors, these innovative projects are displayed to the upcoming batch in the ensuing event.

SNAP CIRCUITS WORKSHOP

Snap circuit product is a tool for opening the exciting world of electronics. Electronics plays an important and increasing role in everyday lives and so some basic knowledge of it is good for all of them. In the workshop, the students had a hands-on session of basic electrical and electronic circuit components. The workshop was conducted for first and second year exclusively. Initially, a brief introduction to the kits was given to the participants. The workshop was broken into two parts. Basic circuits were built using the kits. The participants were asked to make a group of five and visit each circuit in succession where the seniors at each circuit explained them in detail the working of each circuit. Then students were allowed to explore different circuits with the help of the manuals which came with the kits under the supervision of seniors.

GRAPHIC DESIGN WORKSHOP

IEEE SJCE's Editorial Board conducted the Graphic Design Workshop on 16th and 17th September 2019. The main aim of this workshop was to train students and help them learn the basics of designing. Posters are an essential part of any event and hence being well-versed with designing tools is essential.

Topics covered in the workshop :

1.Basics of Photoshop

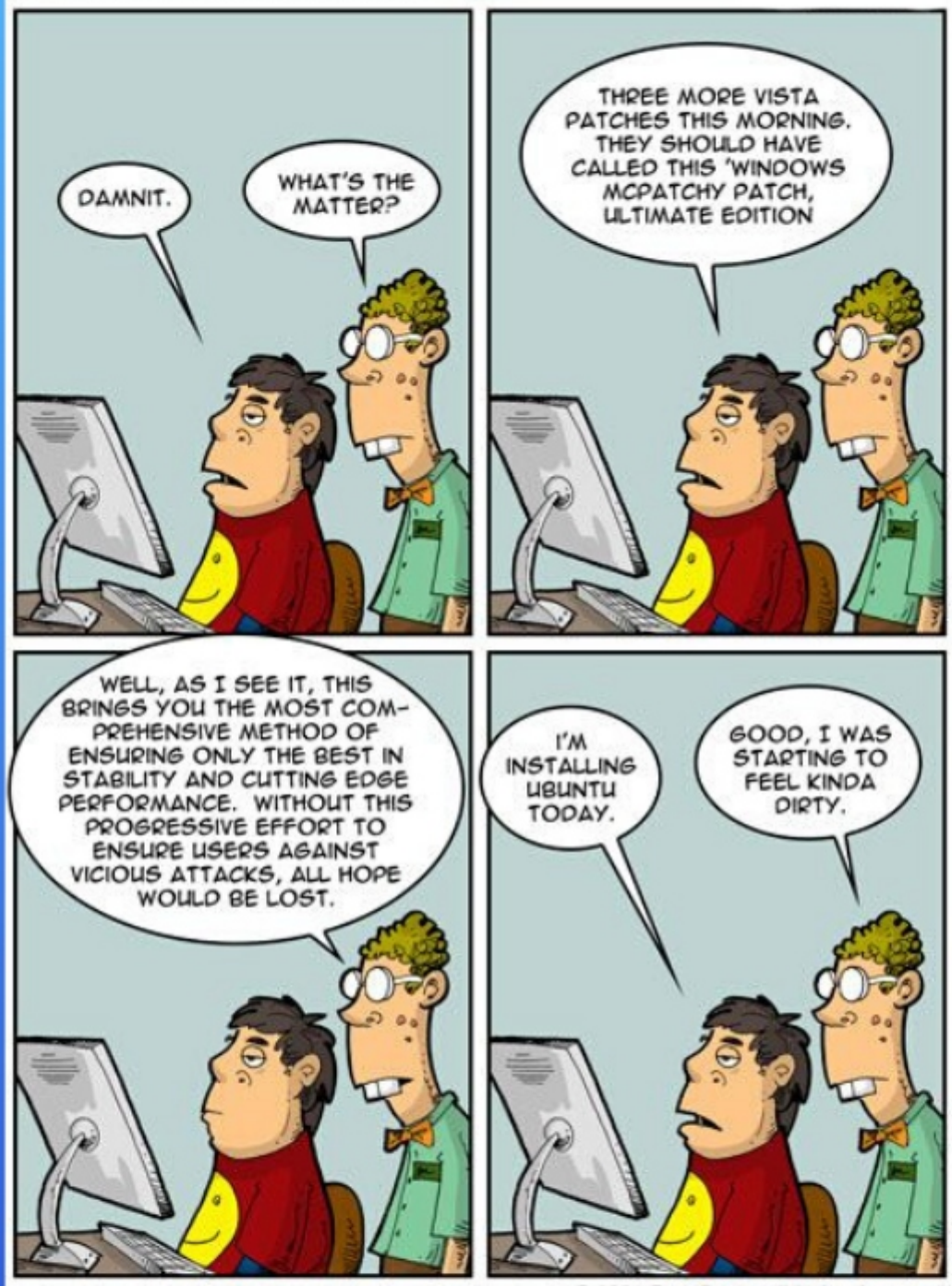
2.GNU's Image Manipulation Program(GIMP) - Pixels and Images, Colour Models, Scaling images, Raster and Vector Graphics, Layers, Channels, Paths, Gradient and perspective tools.

3.Cloud-based simplified graphic design platform – Canva

150 students from various branches attended the workshop. A poster making competition was held for all the participants and the students who secured first and second were recruited into the designing team of Editorial Board of IEEE.

COMICS CORNER

Stack Overflow



OUR TEAM

Shreya A N
Editor-in-Chief

Keerthi V
Editor

Bhargavi P
Designer

Warsa Aliya
Editor

Varun V
Designer

Aditya M G
Designer

For feedback and articles, reach us at - **www.ieeesjce.com**

TUXEDO'19

INQUIZITIVE

A technical and non technical quiz competition

01 INQUIZITIVE

WORKSHOP ON NETWORKING

Electronics and communications enthusiasts? This is tailored for such students.

NETWORKING 02

SHUTTER UP

Obssessed with photography? This is the exact platform.

03 SHUTTER UP

BRAIN CHASE

An online treasure hunt which include puzzles,riddles and questions to crack.

BRAIN CHASE 04

EL DORADO

An offline treasure hunt which will take place in our beautiful campus.

05 EL DORADO

DEBUG AND RUN

A technical event comprising of digital and analog circuits to debug and run on simulation software.

DEBUG & RUN 06