Jurnal Informatika – COMPUTING Volume 07 Nomor 01, BULAN Juni TAHUN 2020: 48 – 73
ISSN: 2656 –3861
e-ISSN: 2722 –2888

DEEP LEARNING DAN TEKNOLOGI BIG DATA UNTUK KEAMANAN IOT

Novianti Indah Putri,S.T[1], Zen Munawar, S.T.,M.Kom[2]

1. Teknik Informatika, Universitas Bale Bandung
2. Manajemen Informatika, Politeknik LP3I Bandung

ABSTRACT

In the field of artificial intelligence (AI), deep learning is a method that belongs to a wider family of machine learning algorithms that work based on the principles of learning. For study, supervised and not supervised, both can be used. In deep learning, a computerized model will perform a series of classification assignments or analysis of specific patterns based on previously learned data. For this reason, a model must first be trained with labeled data sets. Technology has become unavoidable in human life, especially the growth of the Internet of Things (IoT), which enables communication and interaction with various devices. However, IoT is proven to be vulnerable to security breaches. Therefore, it needs to be developed with solutions by creating new technologies or combining existing technologies to overcome security problems. Deep learning, the branch of machine learning has shown promising results in previous studies for detecting security breaches. In addition, IoT devices produce large volumes, variations, and correctness of data. Thus, when big data technology is incorporated, higher performance and better data handling can be achieved. Therefore, we have conducted a comprehensive survey of sophisticated deep learning, IoT security, and big data technology. Furthermore, comparative analysis and the relationship between deep learning, IoT security, and big data technology have also been discussed. Furthermore, thematic classifications have been obtained from a comparative analysis of technical studies from these three domains. Finally, it has identified and discussed the challenges of incorporating deep learning for IoT security using big data technology and has provided direction for future researchers on aspects of IoT security.

Key Word: Deep learning, big data, IoT security

ABSTRAK

Di bidang *artificial intelligence* (AI), *deep learning* adalah metode yang termasuk dalam keluarga yang lebih luas dari algoritma *machine learning* yang bekerja berdasarkan prinsip belajar. Untuk belajar, diawasi dan tidak diawasi, keduanya dapat digunakan. Dalam *deep learning*, model yang terkomputerisasi akan melakukan serangkaian tugas klasifikasi atau analisis pola khusus berdasarkan data yang dipelajari sebelumnya. Untuk itu, seorang model harus dilatih terlebih dahulu dengan set data berlabel. Teknologi telah menjadi tak terhindarkan dalam kehidupan manusia, terutama pertumbuhan Internet of Things (IoT), yang memungkinkan komunikasi dan interaksi dengan berbagai perangkat. Namun, IoT terbukti rentan terhadap pelanggaran keamanan. Karena itu, perlu dikembangkan dengan solusi dengan menciptakan teknologi baru atau menggabungkan teknologi yang sudah ada untuk mengatasi masalah keamanan. *Deep learning*, cabang *machine learning* telah menunjukkan hasil yang menjanjikan dalam studi sebelumnya untuk mendeteksi pelanggaran keamanan. Selain itu, perangkat IoT menghasilkan volume besar, variasi, dan kebenaran data. Dengan demikian, ketika teknologi big data dimasukkan, kinerja yang lebih tinggi dan penanganan data yang lebih baik dapat dicapai. Oleh karena itu, kami telah melakukan survei komprehensif tentang *deep learning* yang canggih, keamanan IoT, dan teknologi *big data*. Selanjutnya, analisis komparatif dan hubungan antara *deep learning*, keamanan IoT, dan teknologi big data juga telah dibahas. Selanjutnya, telah diperoleh klasifikasi tematik dari analisis komparatif studi teknis dari tiga domain tersebut. Akhirnya, telah diidentifikasi dan mendiskusikan tantangan dalam menggabungkan *deep learning* untuk keamanan IoT menggunakan teknologi big data dan telah menyediakan arahan untuk peneliti masa depan pada aspek keamanan IoT.

Kata Kunci: *Deep learning, big data*, keamanan IoT.

ISSN: 2656 –3861
e-ISSN: 2722 –2888

I. PENDAHULUAN

2.1. Latar Belakang

Dasar dari konsep *deep learning* didasarkan pada penelitian jaringan saraf tiruan. Contoh umum dari model yang bekerja dengan arsitektur yang dalam adalah jaringan saraf umpan maju atau *perceptron multilayer* yang terkenal yang terdiri dari banyak lapisan tersembunyi. Di tahun 1980, algoritma lain untuk mempelajari bobot jaringan ini adalah *Back-propagation* [1].

Big data mining menawarkan banyak peluang menarik. Namun, para peneliti dan profesional menghadapi beberapa tantangan ketika mengeksplorasi set *big data* dan ketika mengekstraksi nilai dan pengetahuan dari informasi *mining* tersebut. Kesulitan terjadi di berbagai tingkatan termasuk: pengambilan data, penyimpanan, pencarian, berbagi, analisis, manajemen dan visualisasi. Selain itu, ada masalah keamanan dan privasi terutama dalam aplikasi yang didorong data terdistribusi. Seringkali, kelebihan informasi dan aliran yang didistribusikan melampaui kemampuan untuk memanfaatkan. Faktanya, sementara ukuran *big data* terus meningkat secara eksponensial, kapasitas teknologi saat ini untuk menangani dan mengeksplorasi set *big data*, hanya dalam tingkat petabytes, exabytes, dan zettabytes data yang relatif lebih rendah.

Disamping itu pesatnya pertumbuhan teknologi yang muncul seperti, sensor, smartphone, komunikasi 5G, dan realitas virtual mengarah ke aplikasi inovatif seperti, industri yang terhubung, kota pintar, energi pintar, mobil terhubung, pertanian pintar, kompleks bangunan terhubung, perawatan kesehatan yang terhubung, outlet ritel pintar, dan rantai pasokan cerdas, yang berdampak buruk berkontribusi pada akumulasi sejumlah besar data. Sebuah studi yang dilakukan oleh National Cable and Telecommunications Association (NCTA) memperkirakan bahwa pada tahun 2020, kira-kira 50,1 Miliar perangkat *Internet of Things* (IoT) akan terhubung ke Internet. Pertumbuhan perangkat IoT membuat keamanan perangkat ini dapat diperdebatkan [2] [3] Menurut McAfee (2018), telah ada rentetan serangan *cyber* dan pelanggaran data yang telah menimpa hampir setiap industri sejak 1 Januari 2018. Selanjutnya, banyak dari serangan ini ditargetkan pada perangkat IoT. Meningkatnya penggunaan perangkat IoT mengundang para penjahat dunia maya untuk menargetkannya. Selain itu, prospek interkonektivitas antara perangkat IoT membuat mereka rentan [4]. Selanjutnya, VDC Research Group Inc. juga telah melakukan penelitian untuk menentukan hambatan dalam mengembangkan perangkat yang terhubung. Penelitian telah menunjukkan hal itu 60% dari hambatan terkait dengan persyaratan keamanan dalam mengembangkan perangkat yang terhubung [5]. Selain itu, berdasarkan koleksi Kaspersky Lab, jumlah sampel *malware* untuk Perangkat IoT telah terjadi peningkatan yang cepat dari 3219 sampel untuk tahun 2016 menjadi 1.21588 sampel untuk tahun 2018. Jelas sekali bahwa ada sejumlah besar kerentanan untuk Perangkat IoT [6].

Menurut R. Habeeb [3], banyak organisasi yang menghadapi tantangan terbesar dalam pemantauan ancaman berbasis jaringan, terutama di sektor-sektor berikut: pemerintah, energi, layanan kesehatan, bank, dan pusat penelitian. Selain itu, sektor-sektor ini berinvestasi dalam alat pemantauan keamanan untuk melindungi dan mengamankan infrastruktur mereka. Seperti disebutkan sebelumnya, umumnya, perangkat IoT menghasilkan sejumlah besar data yang mengalir melalui jaringan. Data yang mengalir melalui jaringan berada pada risiko yang mungkin untuk serangan jaringan. Lebih lanjut, penelitian ini berpendapat bahwa alat dan teknik yang ada tidak cukup untuk mendeteksi serangan inovatif yang dipicu oleh penjahat *cyber* karena volume, kecepatan, variasi, dan kebenaran data. Apalagi kapan sejumlah besar data sedang ditangani oleh jaringan, laporan analitik keamanan pada basis mingguan atau bulanan tidak akan cukup untuk mendeteksi dan mengurangi serangan. Lebih jauh lagi, penelitian ini telah menegaskan bahwa teknologi *big data* akan mampu menangani tantangan volume, kecepatan, variasi, dan kebenaran data. Data umumnya dikategorikan sebagai *big data* berdasarkan properti yang terkait dengannya, umumnya dikenal sebagai V *big data* [7]. Teknologi *big data* adalah alat atau teknologi digunakan untuk memproses data ini secara efisien. A.A. Cardenas [8], membahas bahwa perusahaan mengumpulkan keamanan data terkait untuk kepatuhan terhadap peraturan dan analisis forensik post hoc. Selanjutnya, perusahaan besar menghasilkan sekitar 10 hingga 100 miliar acara per hari. Mekanisme yang ada kurang diproses pada skala besar dan analitik *big data* miliki telah digunakan untuk menganalisis dan mengkorelasikan data terkait keamanan secara efisien dan belum pernah terjadi sebelumnya.

Dalam konteks ini, penelitian ini mengusulkan untuk menggunakan *deep learning* dan teknologi *big data* untuk memperkuat keamanan perangkat IoT. Meskipun terlambat, *deep learning* telah mendapatkan pengakuan karena fitur rekayasa non-manual, pra-pelatihan tanpa pengawasan, dan kompresi kemampuan, fitur-fitur ini membuat kemampuan kerja *deep learning* layak bahkan dalam jaringan terbatas sumber daya. Selanjutnya, *deep learning* telah diimplementasikan secara luas karena kemampuan belajar mandiri, potensi untuk menghasilkan hasil yang sangat akurat, dan waktu pemrosesan yang lebih cepat. Ini sangat penting, karena sistem yang terbatas sumber daya dapat

mengalami masalah lain seperti akses di luar memori, bahasa pemrograman yang tidak aman, dan sebagainya [9]. Sebagian besar literatur yang ada secara terpisah berfokus pada *deep learning*, *big data*, dan keamanan IoT. Beberapa penelitian berfokus pada *deep learning* [10][11] atau *big data* [12] [13] untuk keamanan IoT. Sampai saat ini tidak ada studi yang ada yang secara komprehensif meninjau kelayakan menggunakan kedua teknologi ini dalam konteks keamanan IoT. Tabel 1 merangkum sebagian besar studi relevan terkini yang ada dan menyoroti kesenjangan penelitian. Dari Tabel 1, dapat disimpulkan bahwa banyak penelitian telah gagal untuk mempertimbangkan dampak volume, kecepatan, variasi, dan kebenaran data yang dihasilkan oleh perangkat IoT, dibandingkan dengan [3] yang telah menyoroti dampak dalam penelitian tersebut. Oleh karena itu, dimasukkannya teknologi *big data* menjadi wajib untuk mengatasi dampak volume, kecepatan, variasi, dan kebenaran data yang dihasilkan oleh perangkat IoT. Selain itu, jelas terlihat pada Tabel 1 bahwa tidak banyak penelitian telah berfokus pada *deep learning* dan teknologi *big data* untuk keamanan IoT. Penelitian ini dimaksudkan untuk memandu *deep learning, big data*, dan peneliti dan pengembang IoT, kepada siapa keamanan IoT akan menjadi perhatian utama. Kontribusi dari penelitian ini telah dirangkum sebagai berikut : mengidentifikasi, dan menyoroti masalah utama keamanan IoT, memilih lima kasus penggunaan keamanan IoT di mana *deep learning* dan teknologi *big data* bisa menjadi solusi potensial, telah mensurvei penelitian mutakhir yang berfokus pada *deep learning* dan teknologi *big data*, dan keamanan IoT, untuk menentukan penerapan teknis dan batasan dari ketiga domain tersebut di atas, telah mengembangkan taksonomi tematik dengan mengekstraksi informasi berharga dari *state-of-the-art*, telah menganalisis solusi yang ada berdasarkan taksonomi yang diturunkan, telah menyoroti tantangan dan telah mengusulkan pedoman bagi para peneliti di masa depan untuk mendorong keberhasilan penerapan *deep learning* dan teknologi *big data*, dan keamanan IoT.

Namun, penelitian ini membatasi ruang lingkupnya hanya untuk *deep learning* dan tidak membahas tentang algoritma pembelajaran mesin tradisional sehubungan dengan teknologi *big data* dan keamanan IoT. Selain itu, survei ini juga tidak membahas tentang keamanan IoT untuk setiap area aplikasi pintar yang tersedia, melainkan membahas dalam perspektif jaringan dan komunikasi.

II. MOTIVASI DAN KASUS PENGGUNAAN

Pada bagian ini merinci motivasi untuk penelitian dan memberikan beberapa skenario kasus penggunaan yang memotivasi survei *deep learning* dan teknologi *big data* untuk keamanan IoT. Perangkat IoT telah mengalami pertumbuhan pesat dalam beberapa tahun terakhir, yang merupakan masalah besar risiko keamanan yang terkait. Pesatnya pertumbuhan perangkat ini dan ketersediaan teknologi peretasan modern telah memaksa perlunya untuk memastikan bahwa perangkat IoT tidak rentan terhadap pelanggaran keamanan. Namun, sampai sekarang, perangkat IoT telah terbukti terbukti memiliki kerentanan keamanan, seperti ketika perangkat IoT dikompromikan dengan *malware* Mirai dan digunakan untuk menyerang Dyn, penyedia Sistem Nama Domain (DNS). Oleh karena itu, perlu untuk datangnya teknologi baru atau kombinasi dari teknologi yang ada untuk mengamankan perangkat IoT dari penyerang. Persyaratan keamanan IoT seperti kerahasiaan, integritas, ketersediaan, otentikasi, dan kontrol akses membuat perangkat IoT unik dan menantang terutama bagi pengembang untuk menghasilkan sistem IoT canggih yang tahan terhadap serangan berbasis IoT. Studi ini telah dimotivasi oleh fakta bahwa teknologi *big data* mendukung persyaratan keamanan ini dan algoritma *deep learning* telah terbukti efektif dalam deteksi serangan keamanan. Selama bertahun-tahun, *deep learning* telah mendapatkan pengakuan luas di antara para peneliti dan organisasi. Karena kemampuan *deep learning*, ini telah diterapkan di berbagai domain keamanan, seperti [14], [15], dan [16] untuk mengidentifikasi pelanggaran keamanan. Selain itu, *deep learning* telah membuktikan keberhasilannya dalam keamanan IoT, telah terbukti dengan keberhasilan implementasi dalam studi [17], [18], dan [19].

Selain itu, teknologi *big data* juga telah terbukti efektif dalam memproses berbagai jenis data. Studi seperti, [20], [21], dan [22] telah menunjukkan hasil yang menjanjikan. Namun, penelitian terbatas telah dilakukan pada pemrosesan data keamanan IoT dengan teknologi *big data* dan algoritma *deep learning*. Dari analisis kritis dapat mengidentifikasi bahwa hanya dua studi yang menggabungkan *deep learning* dan teknologi *big data* untuk keamanan IoT, yaitu [23] dan [24]. Skenario ini telah memotivasi untuk melakukan penelitian dan akan memotivasi para peneliti di masa depan untuk menggabungkan tiga bidang yang dibahas. Gambar 1 di bawah ini mengilustrasikan kasus penggunaan keamanan IoT dengan penggantian ke teknologi *big data* dan karakteristik *deep learning*.

Gambar 1. Kasus Penggunaan Keamanan IoT

2.1. SirenJack

.

Kerentanan dalam sistem siaran darurat yang diproduksi oleh Acoustic Technology Inc. (ATI) diidentifikasi oleh Balint Seeber yang dijuluki SirenJack, seorang peneliti dari Bastille Security. Sistem memungkinkan siaran paket perintah melalui udara untuk ditangkap, dimodifikasi dan diputar ulang. Cacat itu ditemukan ketika Seeber mengaudit sistem peringatan darurat yang digunakan di seluruh San Francisco [25] [26]. Kasus penggunaan SirenJack adalah jenis deteksi intrusi yang dapat dihindari menggunakan *deep learning* dan teknologi *big data* karena mereka telah menunjukkan hasil yang menjanjikan dalam mendeteksi intrusi.

2.2. Turning Up The Freeze

.

Turning Up The Freeze adalah serangan *Denial-of-Service* (DDoS) yang dilakukan pada sistem kontrol lingkungan di dua gedung apartemen di Finlandia timur. Serangan DDoS menonaktifkan semua sistem kontrol lingkungan di dua apartemen sepenuhnya, yang membuat orang-orang di apartemen kedinginan. Untuk memperbaiki masalah ini, sistem *reboot*. Namun, sistem terjebak dalam loop tanpa akhir [27]. Sistem kontrol lingkungan itu memiliki kemampuan pemrosesan yang mampu mengidentifikasi serangan DDoS dengan mudah menggunakan *deep learning* dan teknologi *big data*. Beberapa rekan peneliti mampu mengidentifikasi serangan DDoS menggunakan *deep learning* dan teknologi *big data*.

2.3. Turn Attack on Dyning

.

Serangan besar dilakukan pada Dyn, penyedia DNS terkemuka pada 21 Oktober 2016. Serangan itu adalah serangan DDoS besar yang membuat sekitar 85 situs web utama seperti Netflix, Twitter, PayPal, dan Sony PlayStation tidak responsif terhadap pengguna. Ini adalah serangkaian tiga serangan, gelombang pertama menyerang pantai Timur, gelombang kedua mempengaruhi California, Midwest, dan Eropa, gelombang ketiga dikurangi oleh Dyn. Serangan diyakini dilakukan oleh sejumlah besar bot IoT yang terinfeksi oleh malware Mirai [28] [29] [30] . Serangan besar ini bisa dikurangi dengan penggunaan teknologi *deep learning* dan teknologi *big data*. Penyedia DNS secara umum menyimpan data log. Data log ini dapat diproses secara efisien oleh teknologi *big data* dan dianalisis menggunakan algoritma *deep learning*, untuk mengidentifikasi semua jenis perilaku anomali. Contoh yang terbukti adalah studi [31], menganalisis perilaku anomali menggunakan teknologi *big data* dan *machine learning*.

2.4. Tangki Air Ikan IoT

.

Di Amerika Utara, peretas telah menggunakan tangki ikan yang terhubung ke internet untuk meretas kasino. Tangki ikan dilengkapi dengan sensor untuk mengatur suhu, pemantauan makanan, dan kebersihan tangki. Peretas menggunakan tangki ikan untuk masuk ke jaringan. Dilaporkan bahwa data bernilai 10 GB ditransmisikan ke perangkat yang berlokasi di Finlandia [32]. Kasus penggunaan ini memberi bukti yang cukup bahwa perangkat IoT dapat digunakan untuk memanipulasi seluruh jaringan. Oleh karena itu, menghentikan penjahat cyber di *firewall* adalah kunci untuk mencegah insiden bencana. Oleh karena itu, pemantauan berkelanjutan aliran data menggunakan *deep learning* dan teknologi *big data* akan memungkinkan deteksi pelanggaran keamanan berbasis IoT pada tahap awal

2.5. Hacked Baby Monitor

.

Monitor bayi dari sebuah keluarga di Ohio diretas oleh peretas yang tidak dikenal. Ketika Adam dan Heather Schreck dan putri mereka yang berumur 10 bulan tertidur, mereka mendengar seorang pria berteriak, “Bangun, sayang! Bangunkan bayi ”dari monitor bayi. Ketika monitor bayi itu diperiksa, keluarga menemukan sudut kamera bergerak sendiri dan suara pria itu menjerit lagi. Ketika Adam Schreck bergegas ke kamar putrinya, sudut kamera berbalik dan menunjuk ke wajahnya dan lelaki itu mulai menjerit-jerit. Orang tua bergegas mencabut kamera. Demikian pula, di Texas monitor bayi nirkabel keluarga diretas dan panggilan bangun serupa terdengar dari monitor bayi [33]. Untuk peretas masuk ke monitor bayi, mereka harus menggunakan jaringan sebagai media.

Jaringan ini dapat diamankan dengan menggabungkan *deep learning* dan teknologi *big data* untuk mendeteksi data atau intrusi apa pun secara real-time. Kasus penggunaan yang dibahas di atas adalah beberapa serangan canggih pada IoT. Namun, jenis serangan terhadap IoT ini terus berkembang dan membutuhkan zaman modern dan sebagian besar solusi baru. Serangan kompleks ini dapat

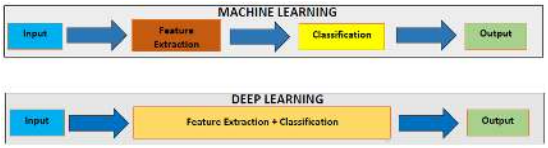
ditangani dengan *deep learning* karena fitur-fiturnya yang membedakan seperti, kemampuan mempelajari fitur yang lebih abstrak, mengurangi kompleksitas pelatihan model, akurasi yang menjanjikan, kemampuan untuk menangani kumpulan *big data*, dan dukungan untuk pembelajaran transfer [34] [35] [36] [37]. Selain itu, teknologi big data dapat memainkan peran penting dalam pemrosesan data IoT, terutama karena volume, kecepatan, dan beragam data yang dihasilkan oleh perangkat IoT. Metodologi yang ada tidak efisien dalam menangani jenis data ini, sehingga teknologi *big data* menjadi suatu keharusan [38]. Selain itu, teknologi *big data* juga mengalami peningkatan kinerja dibandingkan dengan metode tradisional seperti yang diilustrasikan oleh [39] di mana waktu pelatihan jauh lebih sedikit dibandingkan dengan metode pelatihan reguler.

III. KAJIAN PUSTAKA

Bagian ini berisi deskripsi yang komprehensif tentang *deep learning*, teknologi *big data*, dan keamanan IoT. Selain itu, hubungan antara ketiga domain ini telah dibahas, untuk memberikan pengetahuan dasar dan pemetaan hubungan tentang hal ini topik

3.1 Deep Learning

Deep Learning adalah himpunan bagian dari *machine learning* yang memiliki tiga teknik pembelajaran, yaitu pembelajaran yang diawasi, semi-diawasi, dan tanpa pengawasan. Ini terdiri dari banyak lapisan jaringan saraf tiruan. Setiap lapisan berisi beberapa neuron dengan fungsi aktivasi yang dapat digunakan untuk menghasilkan keluaran non-linear. Metodologi ini dikatakan terinspirasi oleh struktur neuron otak manusia [40] [41]. Dalam beberapa tahun terakhir, *Deep learning* telah menarik banyak peneliti dan organisasi, dibandingkan dengan pendekatan *machine learning* tradisional. Para penulis [42] telah membandingkan dalam belajar melawan empat algoritma pembelajaran mesin, seperti, *Support Vector Machine* (SVM), *Decision Trees*, K means, dan Regresi Logistik menggunakan tren Google, dan hasilnya menunjukkan bahwa *deep learning* menjadi lebih populer. Selanjutnya, teknologi ini telah diterapkan dalam berbagai aplikasi AI seperti, pengenalan gambar, pengambilan gambar, mesin pencari dan pencarian informasi, dan pemrosesan bahasa alami. Machine learning dan *deep learning* memiliki empat fase dalam membangun model. Gambar 2 di bawah ini menggambarkan perbedaan antara *machine learning* dan *deep learning.*



Gambar 2: Machine Learning Vs. Deep Learning

Sebagaimana dibahas dalam bagian 2, pembelajaran yang dalam telah mendapatkan pengakuan karena karakteristiknya yang mampu mempelajari fitur yang lebih abstrak, mengurangi kompleksitas pelatihan model, akurasi yang menjanjikan, kemampuan untuk menangani kumpulan data yang besar, dan dukungan untuk pembelajaran transfer [34] [35] [36] [37]. *Deep learning* secara umum telah dijelaskan dalam subbab ini. Dilanjutkan dengan diskusi tentang metodologi khas dan karakteristik *deep learning*.

3.2 Teknologi Big Data

Tidak seperti data tradisional, istilah *big data* mengacu pada set data yang tumbuh besar yang mencakup format heterogen: data terstruktur, tidak terstruktur, dan semi-terstruktur. *Big data* memiliki sifat kompleks yang membutuhkan teknologi canggih dan algoritma canggih. Jadi alat Business Intelligence statis tradisional tidak lagi efisien dalam hal aplikasi *big data.*

Big data dapat dideskripsikan sebagai volume tinggi, kecepatan tinggi, dan variasi tinggi informasi yang menuntut bentuk inovatif dari pemrosesan informasi untuk mendapatkan wawasan dan untuk pengambilan keputusan [43]. Biasanya, *big data* ditandai dengan 6 sifat, umumnya disebut sebagai 6V.

Gambar 3: 6V dari Big Data

Gambar 3 mengilustrasikan 6V, yang merupakan karakteristik dasar dari *big data*, secara umum. Namun, data diklasifikasikan sebagai *big data* selama memenuhi 3V pertama yaitu volume, kecepatan, variasi [44]. Teknologi *big data* dapat digambarkan sebagai alat atau teknologi yang digunakan untuk memproses data secara efisien yang telah diklasifikasikan sebagai

big data. Beberapa teknologi *big data* termasuk, Apache Hadoop [45], Apache Spark [46], Apache Storm [47], Apache Flink [48], Apache Cassandra [49], dan Apache HBase [50].

3.3 Keamanan IoT

Perbedaan utama antara hal keamanan dan serangan keamanan adalah bahwa, hal keamanan adalah hal yang memenuhi semua persyaratan keamanan IAS-oktaf, sedangkan serangan keamanan adalah serangan yang cenderung mengancam setidaknya satu dari IAS-oktaf persyaratan keamanan [1].

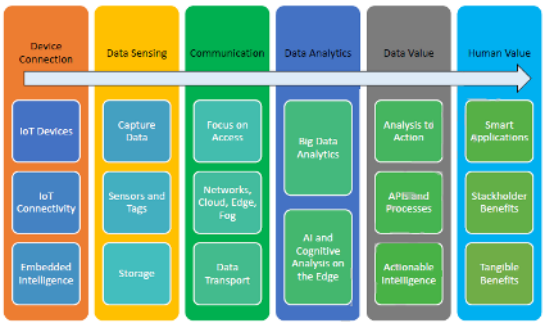
IoT memungkinkan sensor dan perangkat dalam lingkungan yang cerdas untuk berkomunikasi satu sama lain dan memungkinkan berbagi informasi lintas platform. Baru-baru ini IoT telah diadopsi secara luas membangun sistem cerdas seperti, kota pintar, rumah pintar, kantor pintar, gerai ritel pintar, pertanian cerdas, pengelolaan air pintar, transportasi pintar, perawatan kesehatan pintar, dan energi pintar [51] [52] [53]. Karena penggunaan IoT yang luas dalam perangkat seluler, fasilitas transportasi, fasilitas publik, dan peralatan rumah tangga, peralatan ini dapat digunakan untuk akuisisi data di IoT.

Selanjutnya, perangkat yang digunakan dalam berbagai aplikasi yang terhubung ke jaringan IoT dapat dikontrol dari jarak jauh. Perangkat dapat berkomunikasi satu sama lain dan juga dengan perangkat pengendali pusat. Selain itu, ketika digunakan dalam berbagai domain, berbagai data dapat dikumpulkan seperti, data geografis, astronomi, lingkungan, dan logistik [51]. Keamanan IoT dianggap sebagai pengamanan seluruh arsitektur penyebaran IoT dari serangan [54]. Ada berbagai faktor yang perlu dipertimbangkan untuk mengembangkan solusi keamanan IoT. Berikut ini adalah persyaratan keamanan yang harus dipenuhi untuk mengembangkan solusi keamanan IoT. Karena kemampuan luar biasa yang disediakan oleh *deep learning* dan teknologi *big data*, yang dapat digunakan untuk mengidentifikasi kumpulan pelanggaran keamanan yang terkait dengan persyaratan keamanan.

Kerahasiaan memungkinkan informasi dikirimkan secara aman selama semua komunikasi. Ketika informasi dikirimkan tanpa otentikasi atau enkripsi, musuh diberi kesempatan untuk melanggar privasi pemilik [55] [56]. Biasanya, teknologi *big data* terdiri dari transmisi data yang aman dengan menggunakan metodologi enkripsi, sehingga mencegah data untuk dikompromikan oleh musuh [57]. Integritas sistem IoT dapat dikompromikan oleh musuh. Oleh karena itu, integritas menjamin bahwa data yang diterima belum dimanipulasi selama transmisi [58] [56]. Selain itu, Apache Spark, teknologi *big data* memungkinkan dukungan untuk pemeriksaan kualitas data dalam Spark DataFrame [59]. Hal ini memungkinkan pengguna untuk melakukan pemeriksaan integritas data pada sistem IoT. Ketersediaan dalam sistem IoT diperlukan untuk memastikan bahwa pengguna yang sah dapat mengakses sistem dan bahwa akses yang tidak sah ditolak [60], [56]. Salah satu tujuan utama dari teknologi *big data* adalah untuk memastikan kehadirannya di mana-mana bagi pengguna. Selanjutnya, dapat dijalankan pada beberapa node yang memastikan ketersediaan aplikasi yang tinggi [61].

Otentikasi diperlukan untuk memastikan identitas rekan yang berkomunikasi dengan perangkat IoT. Selain itu juga berkaitan dengan pengguna yang valid mendapatkan akses yang tepat untuk tugas-tugas jaringan seperti kontrol perangkat dan jaringan IoT [58], [56]. Selain itu, teknologi *big data* seperti Apache Spark menggabungkan mekanisme otentikasi untuk saluran *Remote Procedure Call* (RPC) [57]. Kontrol akses dalam sistem IoT harus bertindak sebagai sarana untuk memastikan bahwa node yang diotentikasi terbatas untuk mengakses hak keistimewaan dan tidak lebih dari itu [58] [56]. Selain itu, diketahui bahwa teknologi *big data* menyediakan dukungan kontrol akses untuk aplikasinya. Diperlukan filter untuk mencapai hal ini dan setiap aplikasi dapat dilengkapi dengan daftar kontrol aksesnya sendiri [57] Meskipun demikian, *deep learning* tidak secara langsung terkait dengan persyaratan keamanan IoT, pemantauan terus-menerus terhadap jaringan dan komunikasi antara perangkat dan sistem IoT dapat membantu dalam mendeteksi dan memitigasi pelanggaran keamanan pada tahap awal.

Seperti yang dibahas di bagian 3, karakteristik *deep learning* berkontribusi pada identifikasi pelanggaran keamanan, ini karena *deep learning* mampu menangani kumpulan data yang sangat besar, mengklasifikasikan data yang sah dan data anomali pada tingkat akurasi yang lebih tinggi, belajar dari data yang kompleks, dan belajar dari data dengan kecepatan yang jauh lebih cepat.



Gambar 4: Koneksi Perangkat ke Nilai Manusia

di IoT

Gambar 4 mengilustrasikan koneksi ke manfaat perangkat IoT. Bagian di atas telah membahas tentang *deep learning*, teknologi *big data* dan keamanan IoT bersama dengan hubungan di antara ketiganya.

Karena peningkatan yang signifikan dalam penggunaan dan efisiensi pemrosesan data elektronik, hari ini privasi informasi telah menjadi masalah utama. Privasi dalam lingkup IoT dapat diklasifikasikan ke dalam tiga kategori: (i) Kesadaran akan risiko privasi yang dipaksakan oleh hal-hal dan layanan cerdas di sekitar subjek data, (ii) Kontrol individu atas pengumpulan dan pemrosesan informasi pribadi oleh hal-hal cerdas di sekitarnya, (iii) Kesadaran dan kendali atas penggunaan selanjutnya dan penyebaran informasi pribadi oleh entitas-entitas tersebut kepada entitas apa pun di luar lingkup kontrol pribadi subjek [62]. Dalam skenario rumah pintar, *household* subjek atau sekitarnya dapat digambarkan sebagai ruang pribadi subjek dan mungkin cenderung berbeda dari situasi ke situasi. Privasi cenderung bervariasi dalam persepsi dan persyaratan tergantung pada individu sehingga mengarah pada konsepsi informasi pribadi yang tidak jelas. Oleh karena itu, ketika merancang sistem dan layanan baru penilaian yang cermat akan sensitivitas informasi yang terlibat dan persyaratan pengguna terkait harus dipertimbangkan.

IV. *DEEP LEARNING*, TEKNOLOGI *BIG DATA*, DAN KEAMANAN IOT

Pada bagian ini menyoroti dan mengusulkan kaitan *deep learning*, teknologi *big data*, dan keamanan IoT. Pengklasifikasian ke dalam beberapa kategori yaitu, *deep learning*, keamanan IoT, dan teknologi *big data*, dan selanjutnya dikategorikan sebagai arsitektur *deep learning*, kerangka kerja, evaluasi model, Area Aplikasi Keamanan IoT, Serangan keamanan IoT, kumpulan Data, Apache Hadoop, Apache Spark, dan Apache Storm. Karena terbatasnya studi yang telah dilakukan dengan menggabungkan *deep learning*, teknologi *big data*, dan Keamanan IoT, telah diidentifikasi hubungan di antara ketiga domain ini berdasarkan studi eksperimental terkait yang telah menggunakan *deep learning* dengan kombinasi antara keamanan IoT, atau teknologi *big data*, dan keamanan IoT atau teknologi *big data* dengan deteksi serangan keamanan, yang terdiri dari serangan identik seperti itu di ruang IoT.

4.1 Deep Learning

Konsep *Deep Learning* (DL) muncul untuk pertama kalinya pada tahun 2006 sebagai bidang penelitian baru dalam pembelajaran mesin. Ini pertama kali dikenal sebagai pembelajaran hierarkis di awal [63], dan biasanya melibatkan banyak bidang penelitian yang berkaitan dengan pengenalan pola. Pembelajaran mendalam terutama mempertimbangkan dua faktor utama: pemrosesan nonlinear dalam berbagai lapisan atau tahapan dan pembelajaran yang diawasi atau tidak diawasi [64].

Arsitektur *deep learning* umumnya memiliki tiga jenis model pembelajaran, pembelajaran terbimbing, pembelajaran tanpa pengawasan, dan pembelajaran semi-diawasi. Dalam pembelajaran yang diawasi, data yang digunakan untuk melatih arsitektur diberi label penuh, sedangkan dalam pembelajaran yang tidak diawasi, data tidak diberi label dan arsitektur mencoba memunculkan struktur dengan mengekstraksi informasi yang berguna. Dalam model pembelajaran semi-diawasi, dataset pelatihan berisi campuran data yang berlabel dan tidak berlabel, jenis pembelajaran akan ini sia-sia ketika mengekstraksi fitur yang relevan dari data dan membosankan [65]. Selanjutnya, arsitektur *deep learning* dapat dikategorikan menjadi dua jenis, diskriminatif dan generatif. Model diskriminatif umumnya mendukung metode pembelajaran terawasi, sedangkan model generatif mendukung pembelajaran tanpa metode pengawasan [42]. *Autoencoder* (AE): AE adalah jenis Jaringan Syaraf Tiruan (JST) yang mempelajari pengkodean data yang efisien dengan cara yang tidak diawasi [66] [67]. AE terdiri dari *input* dan lapisan *output* yang terhubung menggunakan satu atau lebih lapisan tersembunyi. Secara umum, AE terdiri dari jumlah lapisan *input* dan *output* yang sama. Hal ini bertujuan mengubah *input* menjadi *output* dengan cara sesederhana mungkin, dengan memastikan input tidak terdistorsi sangat banyak [42]. *Recurrent Neural Network* (RNN): RNN dikatakan sebagai perpanjangan dari *Feed Forward Neural Network* (FFNN), yang memanfaatkan informasi sekuensial. RNN mendapatkan nama berulang karena melakukan tugas yang sama untuk setiap elemen dari suatu urutan, di mana *output* tergantung pada perhitungan sebelumnya [68].

Mesin Boltzmann Terbatas (RBM): RBM adalah sejenis JST dengan kemampuan mewakili dan memecahkan masalah yang sulit. RBM terdiri dari dua jenis proses, pembelajaran dan pengujian. Dalam fase pembelajaran, sejumlah besar contoh *input* dan *output* yang diinginkan disajikan untuk menghasilkan struktur RBM di mana aturan umum pemetaan *input* ke *output* dipelajari. Dalam fase pengujian, *output* diproduksi untuk *input* baru oleh RBM, mematuhi aturan umum yang diperoleh dalam fase pembelajaran [69]. *Deep Belief Network* (DBN):

DBN adalah jenis *Deep Neural Network* (DNN) yang terdiri dari beberapa lapisan unit tersembunyi, di mana ada koneksi antara lapisan tetapi tidak dengan unit dari setiap lapisan. Selanjutnya, DBN dapat belajar merekonstruksi inputnya secara probabilistik ketika dilatih dengan contoh-contoh dalam pembelajaran yang tidak diawasi. Selain itu, pada fase post learning, DBN dapat dilatih lebih lanjut dengan supervisi pembelajaran untuk masalah klasifikasi [70] [71] [72].

*Long Short Term Mem*ory (LSTM): LSTM terdiri dari unit khusus yang sering disebut sebagai blok memori dalam lapisan tersembunyi yang berulang. Selanjutnya, blok memori terdiri dari sel-sel memori dengan koneksi-sendiri yang menyimpan keadaan temporal jaringan sebagai tambahan pada unit-unit multiplikasi khusus yang disebut sebagai gerbang, yang mengontrol aliran informasi. Setiap blok memori terdiri dari gerbang *input* dan *output*, di mana gerbang *input* bertanggung jawab untuk aliran aktivasi *input* ke dalam sel memori, dan gerbang *output* bertanggung jawab untuk aliran *output* dari aktivasi sel ke seluruh jaringan [73]. *Convolutional Neural Network* (CNN): CNN adalah jenis JST dalam yang pertama kali diusulkan oleh penulis [74] [75]. CNN menggabungkan algoritma propagasi balik untuk mempelajari bidang reseptif unit sederhana. Selain itu, CNN dicirikan oleh koneksi lokal, pembagian berat badan dan properti pooling lokal. Koneksi lokal dan pembagian berat memungkinkan model ini untuk menemukan pola visual informatif lokal dengan beberapa parameter yang dapat disesuaikan. Properti pooling lokal melengkapi jaringan dengan beberapa terjemahan invarian [76]. Tabel 2 mengklasifikasikan arsitektur yang dibahas di atas berdasarkan kategori, model pembelajaran, dan studi yang telah memanfaatkan arsitektur ini. Selain itu, hubungan dan penerapan arsitektur ini dengan teknologi *big data* dan keamanan IoT telah dibuktikan dengan membuktikan keberhasilan implementasi

Tabel 2: Arsitektur Deep Learning.

Arsitektur	Kategori	Model Pembelajaran	Studi
AE	Generatif	Tidak diawasi	[17] [77]
RNN	Deskriptif	diawasi	[78]
RBM	Generatif	Tidak diawasi dan diawasi	[79] [80] [81]
DBN	Generatif	Tidak diawasi dan diawasi	[82] [83]
LSTM	Deskriptif	Tidak diawasi	[84][19][85][86][87]
CNN	Deskriptif	Tidak diawasi	[88][89][86][87]

Kerangka kerja populer yang biasanya digunakan untuk menerapkan arsitektur *deep learning* adalah sebagai berikut : TensorFlow: TensorFlow adalah kerangka kerja inovatif yang dikembangkan oleh Google, yang menawarkan berbagai perhitungan *deep learning*. TensorFlow secara resmi dirilis pada akhir 2015. Ini termasuk Java, C ++, Go dan Python *Application Programming Interface* (API), dan terutama dirancang untuk perhitungan pada grafik aliran data. Selanjutnya, TensorFlow mendukung komputasi multi-CPU dan multi-GPU dengan ekstensi CUDA dan SYCL. Selain itu, TensorFlow Lite telah dikembangkan untuk memberikan dukungan untuk pembelajaran mesin seluler dan tertanam. Lebih lanjut, TensorFlow Lite menyediakan Android Neural Network API [90].

Theano: Theano adalah pustaka Python *open source*, yang digunakan untuk mengembangkan algoritma yang kompleks melalui ekspresi matematika. Hal ini biasanya digunakan untuk penelitian *machine learning*. Selain itu, telah diterima secara luas di kalangan komunitas *deep learning* karena dukungannya untuk diferensiasi simbolik otomatis dan komputasi yang dipercepat GPU. CUDA digunakan oleh Theano sebagai salah satu pendukung utama untuk perhitungan akselerasi GPU [91]. Caffe: Caffe adalah infrastruktur pelatihan yang banyak digunakan, dikembangkan oleh Berkeley Vision and Learning Center (BVLC) untuk operasi berbasis *deep learning*. DNN disimulasikan sebagai jaringan unit komputasi di Caffe. Unit komputasi umumnya disebut sebagai "lapisan", lapisan ini mengambil data sebagai *input*, melakukan serangkaian operasi, dan meneruskan *output* ke lapisan berikut [92]. PyTorch: PyTorch adalah kerangka *deep learning* berdasarkan python yang bertindak sebagai pengganti NumPy untuk menggunakan kekuatan GPU dan untuk penelitian *deep learning* yang memberikan fleksibilitas dan kecepatan maksimum [93]. PyTorch dikenal luas karena dua fitur utamanya, dukungan akselerasi GPU yang kuat, dan pembangunan jaringan saraf secara dinamis [94].

Microsoft Cognitive Toolkit (CNTK): CNTK adalah kerangka pembelajaran dalam open source untuk Windows dan Linux. Ini digunakan untuk melatih dan mengevaluasi jaringan saraf dalam yang kuat. Microsoft menggunakan toolkit ini untuk model pidato Cortana dan peringkat web. CNTK mendukung berbagai feed forward, convolutional, dan jaringan berulang untuk data ucapan, gambar, dan teks, dan juga kombinasi dari data ini. Selain itu, CNTK dapat meningkatkan skala ke beberapa server GPU dan dirancang untuk mencapai efisiensi [95]. H2O: H2O adalah *machine learning* yang cepat, terukur, dan *open-source* dan kerangka *deep learning* untuk mengembangkan aplikasi pintar. Dukungan untuk algoritma canggih seperti *deep learning*, meningkatkan dan mengantongi elemen membuat H2O lebih disukai untuk aplikasi pintar.

H2O mampu menangani miliaran baris data dalam memori bahkan dalam *cluster* kecil. H2O biasanya dirancang untuk mulai digunakan dalam beberapa menit dan memberikan dukungan untuk Apace Hadoop dan Apache Spark cluster [96].

Deeplearning4j: Deeplearning4j adalah kerangka kerja *open source* untuk komputasi *deep learning* yang dikembangkan oleh tim yang dipimpin oleh Adam Gibson dan didukung oleh organisasi SkyMind. Kerangka kerja ini ditulis dalam Java, Scala, CUDA, C, dan C ++ dan didistribusikan di bawah lisensi Apache 2.0. Selain itu, ini kompatibel dengan Linux, OS X, Windows, dan Android. Deeplearning4j mendukung implementasi semua jaring yang dalam seperti, RBM, DBN, Deep Autoencoder (DAE), dan banyak lagi [97]. Tabel 3 menjelaskan beberapa kerangka kerja yang biasa digunakan untuk *deep learning*, bahasa pemrograman tempat mereka ditulis, versi rilis stabil terbaru, dan tanggal rilis stabil terbaru.

Tabel 3. Framework Deep Learning

Framework	Ditulis dalam	Stabil Terbaru Versi rilis	Stabil Terbaru Tanggal rilis
TensorFlow	Python, C ++, dan CUDA	1.12.0	5 November 2018
Theano	Python, dan CUDA	1.0.4	16 Januari 2019
Caffe	C ++	1	18 April 2017
PyTorch	Python, C ++, dan CUDA		7 Februari 2019
CNTK	C ++	2.7	4 Januari 2019
H2O	Java	3.24.0.3	7 Mei 2019
Deeplearning4j	Java, Scala, CUDA, C, C ++, Python, dan Clojure	0.9.1	13 Agustus 2017

Teknik evaluasi model yang umum digunakan untuk model *deep learning* adalah sebagai berikut. Matriks *Confusion*: Matriks *confusion* adalah ringkasan hasil prediksi model klasifikasi. Matriks *confusion* diperoleh dengan merangkum jumlah total prediksi yang diklasifikasikan secara benar dan salah berdasarkan masing-masing kelas [98]. Perlu untuk mendapatkan nilai-nilai berikut sebelum merancang matriks *confusion*: (a) Benar Positif (TP): Nilai positif sejati merujuk pada jumlah kejadian yang telah diklasifikasikan dengan benar oleh model [99]. (b) Benar Negatif (TN): Nilai negatif sejati adalah jumlah instance negatif yang benar diklasifikasikan oleh model [99]. (c) False Positive (FP): Nilai false positive adalah jumlah instance negatif yang dilabel tidak benar sebagai instance positif [99]. (d) False Negative (FN): Nilai negatif palsu adalah jumlah instance positif yang salah diberi label sebagai instance negatif [99].

Tabel 4 Matriks Confusion.

	Aktual positif	Aktual negatif
Diprediksi positif	TP	FP
Diprediksi negatif	FN	TN

Recall : Recall disebut sebagai sensitivitas atau tingkat positif sejati mengacu pada proporsi contoh positif nyata yang telah diprediksi positif [100]. Penarikan kembali dapat dihitung menggunakan rumus di bawah ini. Recall = TP / (TP + FN). Kekhususan: Kekhususan menggambarkan keefektifan model klasifikasi dalam mengidentifikasi label negatif [101]. Spesifisitas dihitung menggunakan rumus di bawah ini. Spesifisitas = TN / (TN + FP). False Positive Rate (FPR): FPR juga disebut Fall-Out adalah proporsi pada instance negatif yang diklasifikasikan secara tidak benar sebagai instance positif. Dalam istilah yang lebih sederhana, kemungkinan alarm palsu akan dinaikkan [102]. FPR dihitung menggunakan rumus di bawah ini. FalseP ositiveRate = FP / (TN + FP). False Negative Rate (FNR): FNR mengacu pada proporsi sampel yang diklasifikasikan secara tidak benar dengan jumlah sampel positif [103]. FNR dihitung menggunakan rumus : FalseNegativeRate = FN / (TP + FN). Presisi: Presisi adalah proporsi prediksi positif yang positif nyata. Ketepatan diterapkan pada berbagai bidang seperti, pembelajaran mesin, penambangan data, dan pengambilan informasi [100]. Ketepatan dihitung menggunakan rumus di bawah ini: Presisi = TP / (TP + FP).

Pengukuran-F: Pengukuran-f dikatakan sebagai rata-rata harmonik dari presisi dan daya ingat [82]. Pengukuran-f dihitung menggunakan persamaan matematika di bawah ini. F1 = (2 · presisi · recall) / (presisi + recall). Akurasi: Akurasi dapat digambarkan sebagai keefektifan keseluruhan dari model klasifikasi [101]. Rumus yang digunakan untuk perhitungan akurasi adalah sebagai berikut: AC = (TP + TN) / (TP + FP + TN + FN). Matthew's Correlation Coefficient (MCC): MCC adalah teknik yang digunakan untuk

mengukur kualitas klasifikasi biner dan multikelas. Nilai PKS berkisar dari -1 hingga +1, di mana -1 menunjukkan total ketidaksepakatan, 0 menunjukkan predikasi acak dan +1 menunjukkan total perjanjian [104] [105]. MCC dapat dihitung menggunakan rumus di bawah ini: MCC = (TP × TN - FP × FN p) / [(TP + FP) (TP + FN) (TN + FP) (TN + FN)].

Kappa: Kappa juga disebut sebagai Kappa Cohen adalah ukuran inter-reliabilitas. Kappa dikatakan lebih kuat dibandingkan dengan metode perjanjian persen sederhana. Nilai Kappa berkisar 0-1, berikut adalah daftar interpretasi dari Kappa [106]: 0 - 0,20 No Agreement, 0,21 - 0,39 Slight Agreement, 0,40 - 0,59 fair Agreement, 0,60 - 0,79 Subtansial Agreement, 0.80 - 0.90 Almost Perfect. Kappa dihitung menggunakan rumus di bawah ini:

$k = (Po – Pe) / 1 – Pe$

$k = 1 – (1-Po) /(1-Pe)$

4.2 Keamanan IoT

Subbagian ini akan membahas tentang area aplikasi keamanan IoT di mana *deep learning* telah secara jelas diterapkan dengan fokus pada IoT, jenis serangan keamanan pada ruang IoT di mana *deep learning* dapat digunakan untuk mengidentifikasi dan mengurangi serangan-serangan itu, dan akhirnya kumpulan data yang berisi IoT serangan berbasis. Area aplikasi keamanan IoT, area aplikasi keamanan IOT umum di mana *deep learning* telah diterapkan secara jelas telah dibahas di bawah ini.

Deteksi Anomali: Deteksi anomali adalah proses mengidentifikasi anomali. Anomali sering disebut sebagai pola yang tidak mengikuti pola standar. Anomali ini dihasilkan oleh aktivitas abnormal seperti, serangan dunia maya, penipuan kartu kredit, dan banyak lagi. Suatu anomali umumnya dikategorikan ke dalam tiga kategori, yaitu anomali titik, anomali kontekstual, dan anomali kolektif. (a) Anomali titik: Jika *instance* data berbeda dari pola normal dalam dataset, dikatakan anomali titik. (b) anomali kontekstual: Jika dalam konteks tertentu, contoh data berperilaku anomali maka itu disebut anomali kontekstual. (c) Anomali kolektif: Jika sekelompok instance data serupa berperilaku anomali dibandingkan dengan seluruh dataset, mereka dikatakan anomali kolektif [107].

Host Intrusion Detection System (HIDS): HIDS digunakan untuk memantau aktivitas dan karakteristik dari satu host di jaringan untuk aktivitas abnormal apa pun. Umumnya, agen ditempatkan ke host target dalam sistem deteksi intrusi berbasis host. Dalam beberapa kasus, agen dapat digunakan pada perangkat jarak jauh. Sensor dalam intrusi berbasis host, sistem deteksi digunakan sebagai inline atau pasif. Dalam sensor *inline*, lalu lintas jaringan melewati sensor dan kemudian mencapai host target. Sensor pasif memantau replika lalu lintas jaringan nyata [108].

Network Intrusion Detection System (NIDS): NIDS digunakan untuk memantau aliran lalu lintas jaringan. Lapisan jaringan yang berbeda dianalisis oleh NIDS untuk mendeteksi kemungkinan ancaman keamanan [108]. Deteksi *Malware*: Deteksi *malware* adalah proses mengidentifikasi *malware*. Biasanya, ada dua jenis deteksi *malware*, yaitu analisis statis atau dinamis. Dalam analisis statis, *malware* langsung dianalisis dalam bentuk binernya, sedangkan, dalam analisis dinamis, file biner dieksekusi dan kegiatannya dimonitor [109]. Deteksi *Ransomware*: *Ransomware* adalah sebuah tipe *malware* yang mengenkripsi komputer yang terkena dampak dan diminta tebusan untuk dekripsi [110].

Deteksi *Ransomware* adalah proses mengidentifikasi serangan *ransomware*. *Intruder Detection*: Deteksi penyusup adalah proses mengidentifikasi penyusup dengan informasi yang tepat. Penyusup masuk ke dalam 3 kategori berikut: (a) *Masquerader*: Seseorang yang mencoba mendapatkan akses tidak sah ke dalam system (b) *Misfeasor*: Pengguna yang berwenang yang mencoba mengakses fitur istimewa yang dilarang diakses oleh pengguna. (c) Pengguna klandestin: Seseorang yang memperoleh kontrol pengawasan sistem untuk menghindari audit dan kontrol akses atau untuk menekan pengumpulan audit [78].

IoT *Botnet Attack Detection*: Bot adalah perangkat yang terhubung ke infrastruktur protokol umum yang dikendalikan dari jarak jauh. Perangkat dapat dikompromikan dan diubah menjadi bot oleh penyerang. Ketika perangkat IoT bergabung dengan *botnet*, perangkat tersebut dapat digunakan untuk berbagai tujuan, termasuk serangan DDoS [111]. Deteksi serangan *botnet* IoT adalah tindakan mendeteksi serangan berbasis botnet IoT seperti, DDoS. Tabel 5 menunjukkan area *deep learning*, terutama dengan teknologi big data telah diterapkan.

Tabel 5: Area Aplikasi Keamanan IoT

Area Aplikasi Keamanan IoT	Studi
Deteksi Anomali	[81][82]
HIDS	[24]
NIDS	[19][80]45,44 ,29,30]
Deteksi Malware	[89,85,109]

Deteksi Ransomware	[87]
Deteksi Penyusup	[78]
IoT Botnet Attack Detection	[23]

Serangan keamanan IoT, berbagai serangan keamanan IoT berdasarkan pada setiap lapisan adalah sebagai berikut. Serangan Lapisan Persepsi. Lapisan persepsi terdiri dari objek fisik seperti, sensor dan aktuator, node, dan perangkat. Serangan lapisan persepsi mempengaruhi objek fisik dalam infrastruktur IoT. Serangan lapisan persepsi umum telah diuraikan di bawah ini. Botnet seperti Mirai, terdiri dari empat komponen utama: (i) bot adalah malware yang menginfeksi perangkat. Bot terutama bertujuan melakukan dua tugas, yaitu menginfeksi perangkat yang terkonfigurasi dengan salah dan menyerang server target saat menerima perintah dari botmaster, orang yang mengendalikan bot, (ii) antarmuka manajemen terpusat memantau kondisi botnet dan mengatur serangan yang diberikan kepada botmaster melalui server Command & Control (C&C), (iii) loader menyebarkan executable menargetkan berbagai jenis platform seperti, Acorn RISC Machine (ARM), MIPS, dan x86, melalui komunikasi langsung dengan target baru, dan (iv) server laporan digunakan untuk memelihara daftar perangkat di botnet [112] [17].

Sleep derivation attack adalah jenis serangan yang dilakukan pada node dan perangkat sensor bertenaga baterai. Biasanya, perangkat bertenaga baterai mengikuti rutin tidur untuk memperpanjang masa pakainya. *Sleep deprivation attack* bertujuan untuk menjaga agar node dan perangkat tetap terjaga untuk jangka waktu yang lama, yang menghasilkan lebih banyak konsumsi daya baterai dan akhirnya mematikan node dan perangkat [113]. *Node Tampering & Jamming*: Serangan-serangan node Node dipicu ketika seluruh node atau bagian dari perangkat keras node diganti secara fisik. Pengubahan node secara elektronik dapat dicapai dengan menginterogasi node untuk mendapatkan akses dan memanipulasi informasi sensitif, seperti, tabel routing, dan kunci kriptografi bersama. Sedangkan, serangan gangguan simpul adalah ketika seorang penyerang mengganggu frekuensi radio dari node sensor nirkabel, yang memacetkan sinyal dan menunda komunikasi ke node. Asalkan penyerang dapat memacetkan node sensor kunci, layanan IoT dapat ditolak [114].

Eavesdropping: Menguping adalah serangan yang mengancam kerahasiaan pesan. Serangan menguping adalah ketika penyerang sengaja mendengar informasi yang dilewatkan melalui saluran komunikasi pribadi. Frekuensi Radio Identifikasi (RFID) adalah perangkat yang paling rentan untuk menguping jenis serangan [58]. Serangan Lapisan Jaringan. Lapisan jaringan umumnya terdiri dari komponen jaringan seperti, router, jembatan, dan jenis komponen jaringan lainnya. Serangan lapisan jaringan adalah serangan yang ditujukan untuk mengganggu komponen jaringan di ruang IoT.

Man-in-the-Middle (MIM): Dalam serangan MIM, seorang penyerang memiliki kendali penuh atas saluran komunikasi antara dua entitas yang sah. Lebih lanjut, penyerang tidak terbatas pada membaca pesan, tetapi untuk mengubah, menghapus, dan memasukkan pesan ke dalam saluran komunikasi [115]. (a) *Address Resolution Protocol* (ARP) *Cache Poisoning*: Protokol ARP menargetkan resolusi alamat MAC dari host yang diberikan IP-nya. Ini dicapai dengan mengirimkan permintaan paket ARP di jaringan. Keracunan cache ARP juga disebut sebagai ARP spoofing, routing racun ARP adalah proses pemalsuan Paket ARP yang memungkinkan peniruan host lain di jaringan [116]. (b) DNS *Spoofin*g: DNS memetakan nama simbolik ke alamat IP. *Spoofing* DNS kadang-kadang disebut sebagai keracunan cache DNS, memengaruhi resolver DNS dengan menyimpan informasi pemetaan berbahaya antara nama simbolis dan alamat IP. Server DNS dapat diracuni oleh penyerang dengan mengkompromikan server DNS resmi atau memalsukan respons terhadap permintaan DNS rekursif [117]. (c) Pembajakan Sesi: Serangan pembajakan sesi adalah tindakan jahat dari penyerang yang berhasil mengamankan pengidentifikasi sesi pengguna, yang memungkinkan penyerang untuk mentransfer sesi ke sistemnya sendiri [118].

Denial of Service (DoS) / DDoS: DoS adalah jenis serangan jahat yang bertujuan dalam menghabiskan sumber daya atau bandwidth pengguna asli. DDoS adalah varian dari DoS yang mirip dengan serangan DoS tetapi melibatkan berbagai node yang dikompromikan. [119]. (a) *User Datagram Protocol* (UDP) *Flood*: UDP *Flood* adalah serangan banjir di mana banyak datagram UDP dihasilkan secara khas oleh bot. Datagram UDP ini membanjiri berbagai bagian jaringan dan memadatkan seluruh jaringan [120]. (b) *Internet Control Message Protocol* (ICMP) Flood: ICMP banjir disebut sebagai ping banjir di mana paket ICMP *Echo Request* (ping) yang berkelanjutan dikirim ke tuan rumah secepat mungkin tanpa menunggu balasan. Ini akan menghabiskan sumber daya komunikasi yang masuk dan keluar ketika tuan rumah mencoba untuk membalas ping [119]. (c) *Flood* SYN: Dalam serangan banjir SYN

penyerang mengirimkan sejumlah besar paket SYN *Transmission Control Protocol* (TCP) ke target. Ini memaksa target untuk menggunakan sumber daya terbatas seperti, CPU, *bandwidth*, dan memori untuk membalas SYN. Kecepatan serangan yang tinggi akan menyebabkan serangan DoS dan akhirnya tidak dapat melayani pengguna asli [121]. (d) *Ping of Death* adalah serangan, di mana penyerang mengirimkan ping berukuran sangat besar ke target dengan maksud untuk menjatuhkan target. Banyak sistem operasi cenderung macet ketika ukuran ping telah terlampaui [122]. (e) Slowloris adalah serangan DDoS, di mana beberapa permintaan *HyperText Transfer Protocol* (HTTP) dibuka dan dimanipulasi secara bersamaan antara penyerang dan target. Slowloris mampu menciutkan aplikasi dengan menggunakan lalu lintas dan penyerang minimal [123]. (f) *Network Time Protocol* (NTP) *Amplification*: NTP *Amplification attack* adalah jenis serangan DDoS volumetrik berbasis refleksi di mana NTP dieksploitasi oleh penyerang untuk membanjiri lalu lintas UDP yang diamplifikasi ke host. Oleh karena itu, ini mempengaruhi host dan infrastruktur sekitarnya yang menyebabkan lalu lintas reguler tidak dapat diakses oleh sumber daya [124].

Routing Attacks: Dalam serangan routing, node berbahaya meluncurkan jenis serangan routing untuk mengganggu operasi routing atau untuk melakukan serangan DoS [124]. (a) Serangan Sybil: Selama serangan Sybil, node jahat merusak sistem perutean, dan mengakses informasi yang diblokir oleh node, atau jaringan dipartisi. Serangan ini dieksekusi oleh penyerang tunggal yang menciptakan banyak identitas palsu dan berpura-pura menjadi banyak dalam jaringan peer-to-peer (P-2-P) [125]. (b) *Sinkhole Attack*: Serangan *Sinkhole* dilakukan dengan terdiri dari node yang mencoba untuk menarik lalu lintas sebanyak mungkin dari area tertentu, dengan membuat dirinya terlihat menarik untuk node sekitarnya berdasarkan pada metrik routing. Oleh karena itu, simpul jahat menarik semua lalu lintas dari stasiun pangkalan. Ini kemudian memberikan penyerang untuk melakukan serangan lebih lanjut pada sistem [133]. (c) *Selective Forwarding Attack*: Serangan penerusan selektif mampu melakukan serangan DoS di mana node jahat meneruskan paket secara selektif. Tujuan dari serangan ini umumnya adalah untuk mengganggu jalur routing. Namun demikian, ini dapat digunakan untuk menyaring protokol apa pun [134]. (d) Serangan Wormhole: Tujuan dari serangan wormhole adalah untuk mengganggu topologi jaringan dan arus lalu lintas. Serangan lubang cacing terjadi ketika sebuah node jahat mengirim pesan di antara dua bagian jaringan yang berbeda melalui tautan kecepatan tinggi [135, 136]. (e) *Hello Flood* adalah salah satu serangan utama di lapisan jaringan. Serangan hello banjir memungkinkan penyerang untuk memaksa node konvensional kehilangan daya dengan memaksa mereka untuk mengirimkan paket halo besar dengan daya yang sangat tinggi [137].

Serangan Middleware: Dalam infrastruktur IoT middleware terdiri dari komponen-komponen seperti cloud. Serangan middleware secara langsung melibatkan aktivitas jahat pada komponen middleware dari infrastruktur IoT. (a) Berbasis Cloud: Dalam serangan berbasis cloud, penyerang langsung menyerang platform cloud karena berbagai alasan, seperti pencurian informasi, serangan banjir, dan sebagainya. Serangan berbasis cloud yang umum meliputi: . *Cloud Malware Injection*: Selama serangan injeksi malware cloud, seorang penyerang mendapatkan akses ke data korban di cloud dan mengunggah salinan jahat dari instance layanan korban, oleh karena itu memungkinkan layanan korban diproses dalam instance berbahaya [138].

Cloud Flooding Attack: Serangan cloud flooding memungkinkan penyerang mengirim sejumlah besar paket dari host yang tidak bersalah dalam jaringan untuk membanjiri korban. Paket besar ini dapat berupa kombinasi atau banyak TCP, UDP, dan ICMP. Selanjutnya, jenis serangan ini dapat memengaruhi kemampuan layanan untuk melayani pengguna yang berwenang. Selain itu, penggunaan cloud dapat meningkat karena tidak memiliki kemampuan mengidentifikasi lalu lintas yang sah dan menyerang [139]. (b) Serangan Otentikasi: Serangan berbasis otentikasi digunakan untuk mengeksploitasi proses otentikasi yang digunakan untuk memverifikasi pengguna, layanan, atau aplikasi [140].

Brute Force: Serangan brute-force membuat penyerang mendapatkan akses dengan memasukkan berbagai kredensial login dengan harapan menebak kredensial dengan benar. Penyerang memasuki berbagai kemungkinan kata sandi sampai kata sandi yang tepat ditemukan [141]. *Kamus Attack*: Serangan kamus juga disebut sebagai serangan mempertanyakan kata sandi adalah ketika seorang penyerang telah membangun database dengan kemungkinan kata sandi. Penyerang mengeksekusi ini dengan menguping di saluran dan mencatat transkrip. Setelah itu, kata sandi dicoba untuk dibuat agar sesuai dengan yang direkam. Jika kecocokan telah ditemukan, maka penyerang telah berhasil memperoleh kata sandi [142].

Replay Attack: Serangan replay memungkinkan penyerang mencegat dan menangkap komunikasi atau tindakan digital dan menggunakannya pada titik waktu berikutnya. memungkinkan penyerang menggunakan informasi orang lain untuk menyamar sebagai

orang itu [143]. (c) *Signature Wrapping Attack*: Serangan pembungkus tanda tangan memungkinkan penyerang muncul sebagai pengguna yang sah dan melakukan permintaan layanan web sewenang-wenang. Ini dicapai dengan menyuntikkan elemen jahat ke dalam struktur pesan, ini memastikan tanda tangan yang valid untuk elemen yang sah dan pemrosesan elemen jahat dalam logika aplikasi [144].

Serangan Lapisan Aplikasi. : Lapisan aplikasi adalah aplikasi itu sendiri, seperti rumah pintar, kota pintar, dan smart grid. Serangan lapisan aplikasi terkait dengan pelanggaran keamanan aplikasi IoT. Serangan layer aplikasi terkemuka telah dijelaskan di bawah ini. (a) *Malware* adalah jenis serangan, di mana kode yang dapat dieksekusi digunakan oleh penyerang untuk mengganggu perangkat di jaringan. Ini memungkinkan penyerang untuk mendapatkan akses tidak sah atau mencuri informasi sensitif. Dalam jaringan IoT, penyerang dapat memanfaatkan kelemahan firmware dan mampu mengganggu seluruh arsitektur IoT [145, 146]. (b) Serangan *Phishing*: *Phishing* adalah jenis serangan yang bertujuan untuk mengekstrak informasi sensitif seperti, nama pengguna, dan kata sandi dari pengguna dengan tampaknya menjadi entitas yang dapat dipercaya. Informasi sensitif dapat digunakan kemudian oleh penjahat cyber untuk membahayakan pengguna atau sistem [147].

Phear Tombak: Phear tombak ditargetkan khusus pada individu dan organisasi tertentu, bukan pengguna acak. Penyerang umumnya meningkatkan pengetahuannya tentang target dan pengaturan. Penyerang kemudian dapat mengirim pesan dengan berpura-pura sebagai entitas yang sah [148]. Kloning Phishing: Kloning phishing adalah ketika email yang sah yang dikirim sebelumnya dikloning ke email berbahaya yang umumnya berisi tautan ke situs web phisher [148].

Perburuan paus: Perburuan paus mirip dengan phishing tombak kecuali bahwa itu terutama ditargetkan pada eksekutif senior perusahaan dan pejabat pemerintah [148].

(c) Serangan Injeksi Kode: Serangan injeksi kode berfokus pada mendepositkan kode yang dapat dieksekusi berbahaya (kode mesin) ke ruang alamat dari proses korban, dan kemudian memberi wewenang kontrol ke kode ini [149].

Structured Query Language (SQL) Injection: SQL injection dijalankan pernyataan SQL database berbahaya dengan mengambil keuntungan dari validasi aliran data dari pengguna ke database [150]. Injeksi Skrip: Selama injeksi skrip atau Cross-Site Scripting (XSS), skrip jahat, yang umumnya ditulis dalam JavaScript disuntikkan ke dalam konten situs web. Skrip berbahaya mampu membocorkan informasi sensitif dari situs [151]. Injeksi Shell: Serangan injeksi shell kadang-kadang disebut sebagai serangan injeksi perintah menyuntikkan perintah jahat ke dalam sistem untuk melakukan aktivitas jahat [152].

Dataset yang sering digunakan untuk analisis eksperimental pada *deep learning*, teknologi big data dan / atau untuk keamanan IoT atau keamanan jaringan adalah sebagai berikut. UNSW-NB15: Dataset UNSW-NB15 dikembangkan pada tahun 2015, yang terdiri dari kombinasi data serangan normal yang disintesis normal modern dan kontemporer. Ini adalah dataset berlabel dan terdiri dari total 47 fitur. Selanjutnya, dataset ini terdiri dari 9 tipe serangan, yaitu fuzzes, analisis, backdoors, DoS, exploit, generic, reconnaissance shellcode, dan jenis serangan worm [157] NSL-KDD: Dataset ini merupakan perpanjangan dari dataset KDDCUP99, di mana catatan yang dipilih diekstraksi dari seluruh dataset KDDCUP99. Dalam penelitian [158], penulis telah menegaskan bahwa dataset KDDCUP99 sangat memengaruhi kinerja sistem yang dievaluasi dan menghasilkan buruknya evaluasi teknik deteksi anomali. Karena itu, mereka telah mengusulkan NSL-KDD, yang tidak termasuk catatan redundan di set kereta, set tes yang diusulkan tidak mengandung catatan duplikat, di tangan di setiap tingkat kesulitan jumlah catatan yang dipilih berbanding terbalik dengan persentase catatan dalam dataset KDDCUP99 , catatan set kereta dan tes masuk akal. NSL-KDD dataset terdiri dari empat jenis serangan, yaitu DoS, User to Root (U2R), Remote to Local (R2L), dan serangan Probe. KDDCUP99: Kumpulan data KDDCUP99 dibuat oleh penulis studi [159] berdasarkan pada program evaluasi IDS DARPA'98 [160]. Selain itu, dataset ini banyak digunakan di kalangan peneliti untuk evaluasi pendekatan deteksi anomali.

Dataset DARPA'98 adalah sekitar 4 gigabytes data tcpdump dari 7 minggu lalu lintas jaringan. Selanjutnya, data pelatihan dataset terdiri dari sekitar 4.900.000 koneksi vektor tunggal di mana masing-masing terdiri dari 41 fitur, yang dilabeli sebagai serangan atau data normal. Dataset ini terdiri dari 4 jenis serangan, serangan DoS, U2R, R2L, dan Probe [158].

WSN-DS: Kumpulan data WSN-DS dibuat oleh [161] berdasarkan lalu lintas jaringan dalam node sensor nirkabel. Dataset ini terdiri dari total 26 fitur berlabel. Selain itu, WSN-DS terdiri dari 4 jenis serangan berbasis DoS, yaitu serangan lubang hitam, serangan lubang abu, serangan banjir, dan serangan penjadwalan [161].

IoTPOT: Kumpulan data IoTPOT dikembangkan oleh [162] yang terdiri dari lalu lintas jaringan IoT. Dataset ini terdiri dari lalu lintas jaringan berbasis normal dan malware, terutama digunakan dalam serangan berbasis

DDoS. Dataset diklasifikasikan berdasarkan 5 keluarga malware, yaitu ZORRO, GAYFGT, nttpd, KOS, dan * .sh [162]. Kyoto: Dataset Kyoto dibangun pada 2006 untuk penelitian Sistem Deteksi Intrusi (IDS).

Dataset ini dibangun berdasarkan 3 tahun dari data lalu lintas jaringan nyata. Selanjutnya, 14 fitur yang berasal dari KDDCUP99 dan 10 fitur tambahan telah dimasukkan dalam dataset ini. Selanjutnya, data honeypot mereka terdiri dari total 50.033.015 normal sesi dan 43.043.255 sesi serangan. Selain itu, dibahas tentang 3 jenis serangan, exploit, shellcodes, dan malware [163].

CICIDS2017: Dataset CICIDS2017 dibuat oleh Canadian Institute for Cybersecurity (CIC) pada 2017. Ini berisi data lalu lintas jaringan jinak dan serangan jaringan nyata. Dataset ini terdiri dari 225.746 catatan dengan total 80 fitur. Selain itu, kumpulan data ini terdiri dari jenis serangan Brute Force, Web, DoS, Botnet, dan DDoS [164].

Set Data Deteksi Intrusi Coburg (CIDDS) -001: CIDDS-001 adalah dataset berbasis aliran berlabel yang dikembangkan untuk evaluasi NIDS berbasis anomali. Dataset terdiri dari data lalu lintas normal dan serangan yang dikumpulkan selama periode empat minggu. Selanjutnya, dataset ini terdiri dari 14 fitur dan 4 jenis serangan seperti, DoS, PortScan, Brute memaksa, dan memindai Ping [165].

4.3 Teknologi Big Data

Subbagian ini membahas teknologi big data penting yang ada yang diterapkan dalam konteks *deep learning* untuk keamanan IoT atau keamanan jaringan. Selain itu, teknologi big data, platform pengembangan mereka, versi stabil terbaru, tanggal rilis stabil terbaru, dan beberapa studi yang telah menerapkan teknologi big data dengan *deep learning* dan / atau untuk keamanan IoT atau keamanan jaringan untuk permintaan yang membutuhkan sekitar 5 atau lebih siklus.

Apache Hadoop adalah alat pemrosesan batch yang menyediakan skalabilitas dan toleransi kesalahan. Hadoop mendukung petabyte data dan memungkinkan aplikasi dijalankan pada banyak node. Selain itu, data log dipecah menjadi blok dan dikirim ke node di Hadoop gugus. Selain itu, Hadoop populer karena kemampuan pengambilan cepat, pencarian data log, skalabilitas, penyisipan data yang lebih cepat, dan toleransi kesalahan [126]. Apache Spark dikembangkan sebagai model terpadu untuk pemrosesan data terdistribusi oleh University of California, Berkely pada tahun 2009. Spark memperluas model MapReduce dengan abstraksi berbagi data yang disebut Resilient Distributed Dataset (RDD). Menggunakan ekstensi ini, Spark dapat menangkap dan memproses beban kerja seperti, SQL, streaming, pembelajaran mesin, dan pemrosesan grafik [46].

Apache Storm adalah sistem perhitungan real-time open source. Storm memungkinkan pemrosesan aliran data secara praktis. Lebih lanjut, ia mampu memproses jutaan tupel per detik per node. Storm cepat, scalable, toleran terhadap kesalahan, dan ramah pengguna. Bahkan, storm menyediakan kemampuan untuk menggabungkan basis data dalam pemrosesan [47].

V. DEEP LEARNING UNTUK KEAMANAN IOT DENGAN TEKNOLOGI BIG DATA

Bagian ini terdiri dari tiga subbagian. Subbagian pertama menyajikan wawasan teknik canggih dalam kasus di mana *deep learning* telah diterapkan untuk keamanan IoT. Subbagian kedua merinci tentang penerapan *deep learning* bersama dengan teknologi big data. Akhirnya, tinjauan komprehensif *deep learning*, teknologi big data dan keamanan IoT telah disajikan.

5.1. Deep Learning dan Keamanan IoT

Subbagian ini membahas teknik canggih yang digunakan untuk keamanan IoT menggunakan teknik *deep learning*. IoT telah mendapatkan banyak perhatian sehingga militer pun menggunakan IoT. *Internet of Battlefield Things* (IoBT) disebut sebagai penggunaan IoT untuk operasi militer dan aplikasi pertahanan. Para penulis penelitian [89] telah mengidentifikasi bahwa injeksi malware adalah serangan yang paling umum. Selanjutnya, mereka telah mengusulkan Eigenspace yang mendalam pendekatan pembelajaran untuk mendeteksi malware IoBT melalui urutan *Operational Codes* (OpCode) perangkat. OpCodes ditransmutasikan ke dalam ruang vektor dan pembelajaran Eigenspace yang mendalam. Pendekatan ini digunakan untuk mengklasifikasikan aplikasi jinak dan jahat. Selain itu, mereka telah mengevaluasi keberlanjutan pendekatan yang diusulkan terhadap serangan penyisipan kode sampah. Telah mengevaluasi model mereka berdasarkan empat metrik evaluasi, yaitu akurasi, ketepatan, daya ingat, dan ukuran-f. Selain itu, mereka telah membandingkan dua studi serupa lainnya berdasarkan metrik. Relatif, pendekatan yang diusulkan mereka telah mencapai akurasi yang lebih baik dari 99,68%, presisi 98,59%, penarikan kembali 98,37%, dan f-ukur 98,48%.

Selanjutnya, model yang diusulkan telah mampu mengurangi serangan penyisipan kode sampah [89]. Namun demikian, dataset yang digunakan dalam penelitian ini adalah dataset yang dibuat sendiri. Kualitas dan validitas data masih bisa diperdebatkan. Selain itu, sejumlah sampel malware disertakan dalam dataset.

Perangkat IoT yang lebih mudah dikompromikan dibandingkan dengan komputer desktop telah menyebabkan peningkatan serangan botnet IoT. Untuk mengurangi ancaman ini, Meidan [17] telah mengusulkan penggunaan DAE untuk mendeteksi lalu lintas jaringan yang tidak normal dari perangkat IoT yang dikompromikan.

Deep Learning telah diterapkan pada snapshot perilaku yang diekstraksi dari jaringan. Untuk mengevaluasi model mereka, mereka telah menginfeksi sembilan perangkat IoT komersial dengan botnet Mirai dan BASHLITE. Model dievaluasi berdasarkan True Positive Rate (TPR), False Positive Rate (FPR), dan waktu deteksi serangan. Hasil TPR yang diterima adalah 100%, sedangkan rata-rata FPR adalah 0,007 + - 0,01 untuk model yang diusulkan. Selanjutnya, Model mengambil 174 + - 212 milidetik untuk mendeteksi serangan. Namun, model hanya dievaluasi berdasarkan dua botnet, yaitu botnet Mirai dan botnet BASHLITE. Selain itu, model yang diusulkan hanya dibandingkan dengan tiga model pembelajaran mesin. Perbandingan model *deep learning* lainnya akan lebih memperjelas keakuratan model. *Deep Learning* dengan kemampuannya seperti, kemampuan ekstraksi fitur tingkat tinggi, kemampuan mandiri, dan kemampuan kompresi menjadikannya penemuan pola tersembunyi yang ideal yang membantu dalam membedakan serangan dari lalu lintas yang tidak berbahaya.

Oleh karena itu, penelitian [18] mengusulkan *deep learning* Pendekatan berdasarkan Stochastic Gradient Descent (SGD), yang memungkinkan deteksi serangan di IoT sosial. Model telah dievaluasi berdasarkan akurasi, presisi, penarikan kembali, f1-measure, tingkat deteksi, dan False Alarm Rate (FAR). Hasilnya menunjukkan bahwa model yang dalam telah mengungguli model yang dangkal di setiap aspek evaluasi. Selain itu, dibahas bahwa pembelajaran yang dalam menunjukkan kinerja yang lebih baik dibandingkan dengan pembelajaran mesin tradisional model. Sebaliknya, serangan yang dievaluasi terbatas, seperti serangan DoS, Probe, R2L, dan U2R.

Demikian juga, hanya satu set data yang telah digunakan untuk mengevaluasi model, yaitu set data NSL-KDD. Selain itu, penulis penelitian [19] telah mengusulkan teknik *deep learning* yang memungkinkan deteksi intrusi dalam jaringan IoT menggunakan Bi-directional LSTM Recurrent Neural Network (BLSTM RNN). Model ini telah dievaluasi menggunakan tujuh metrik, yaitu akurasi, presisi, daya ingat, skor f1, laju salah perhitungan, FAR, dan waktu deteksi.

Model yang diusulkan mampu mencapai akurasi tinggi 95,7%. Di sisi lain, model yang diusulkan telah dievaluasi pada satu dataset. Juga, model tidak dibandingkan dengan model serupa dalam hal evaluasi. Selanjutnya, dalam penelitian [85] penulis telah mengusulkan model *deep learning* menggunakan LSTM untuk mendeteksi malware di IoT berdasarkan urutan OpCodes. Model telah dievaluasi berdasarkan akurasi, TP, FP, TN, dan FN.

Akurasi yang diperoleh adalah 98% pada malware baru, malware tidak ada dalam data pelatihan. Sebaliknya, dataset yang ditiru telah digunakan dalam penelitian ini. Selain itu, ada sampel / file dataset terbatas, dengan total 180 malwares dan 271 file jinak. Selain itu, penulis studi [80] telah memperkenalkan kerangka kerja untuk IoT berdasarkan Software Defined Networking (SDN). Mereka terutama berfokus pada aplikasi IoT, di mana keamanan sangat penting, seperti kota pintar. Mereka telah menggunakan RBM untuk menyebarkan IDS untuk mendeteksi anomali. Mereka telah membandingkan pendekatan yang diusulkan dengan algoritma pembelajaran mesin dan telah mengevaluasinya berdasarkan delapan metrik, TP, FP, TN, FN, presisi, daya ingat, False Discovery Rate (FDR), dan False Negative Rate (FNR). Mereka mampu mencapai tingkat presisi lebih dari 94%. Namun demikian, mereka telah memilih dataset KDD99, ini adalah dataset yang sudah ketinggalan zaman yang berisi serangan tahun 1999. Termasuk set data terbaru yang berisi serangan modern akan meningkatkan keandalan model. Karena kenyataan bahwa dataset ini sudah usang, mereka hanya berisi jenis serangan terbatas seperti, DoS, Probe, Reconnaissance, R2L, dan U2R. Dalam studi [40] penulis telah membahas bahwa aplikasi IoT menghadapi masalah keamanan utama di kerahasiaan, integritas, privasi, dan ketersediaan. Oleh karena itu, mereka telah mengusulkan model untuk deteksi serangan siber di lingkungan IoT.

Sebanyak empat metrik evaluasi telah digunakan untuk evaluasi model, yang mencakup ketepatan, ketepatan, daya ingat, dan waktu deteksi. Hasil mengungkapkan kekokohan akurasi dan penghematan waktu yang signifikan. Namun, akurasi model telah di atas 95% untuk dataset NSL-KDD sedangkan untuk UNSW-NB15 semua model telah mencapai akurasi kurang dari 95%. Selanjutnya, waktu juga meningkat dalam UNSW-NB15 dibandingkan dengan dataset NSL-KDD. NSL-KDD adalah perpanjangan dari dataset KDD99 dengan modifikasi tertentu yang dibuat.

Padahal, dataset UNSW-NB15 adalah dataset yang lebih baru yang berisi serangan modern. Hal ini dapat dilihat dari hasil bahwa model berkinerja lebih baik pada dataset yang lebih lama dan penurunan kinerja pada dataset terbaru. Selain itu, penulis dalam penelitian ini

[86] telah mengusulkan dan menerapkan empat algoritma *deep learning* dan membandingkannya dengan algoritma pembelajaran mesin tradisional. Lebih lanjut, mereka telah mengidentifikasi bahwa algoritma hybrid LSTM + CNN telah mengungguli semua algoritma lainnya dibandingkan dengan algoritma pembelajaran yang dalam dan pembelajaran mesin, dengan yang menakjubkan akurasi 97,16%. Secara komparatif, semua model *deep learning* telah mengungguli model pembelajaran mesin. Sebaliknya, dataset dimanipulasi untuk menyeimbangkan data karena terdiri dari data yang sangat tidak seimbang. Selain itu, metrik evaluasi model terbatas adalah digunakan seperti, akurasi, presisi, dan daya ingat.

Selanjutnya, metrik evaluasi seperti, f-ukur, MCC, dan TPR, mungkin memiliki nilai tambah bagi model. Selain itu, penulis penelitian [127] telah mengusulkan pendekatan *deep learning* dengan Dense Random Neural Network (DRNN) untuk memprediksi kemungkinan serangan jaringan yang sedang berlangsung berdasarkan paket capture.

Metodologi mereka terutama berfokus pada deteksi online serangan jaringan terhadap gateway IoT. Mereka telah menemukan bahwa hasil yang mereka peroleh sebanding dengan hasil dari detektor ambang sederhana. Namun demikian, studi mereka hanya berfokus pada jenis serangan terbatas di ruang IoT seperti, *flood* UDP, TCP SYN, serangan kurang tidur, serangan rentetan, dan serangan siaran. Selanjutnya, hasilnya belum dibandingkan dengan algoritma lain atau dengan penelitian serupa. *Ransomware*, adalah *malware* yang berkembang pesat yang telah mempengaruhi berbagai industri di berbagai negara. Oleh karena itu, studi [87] mengusulkan model yang menggunakan LSTM dan CNN untuk membedakan *ransomware* dan *goodware* dalam jaringan. Metrik evaluasi yang digunakan untuk model adalah f-ukur, TPR, FPR, dan MCC. Diklaim bahwa model tersebut memperoleh ukuran-f 99,6% dengan TPR 97,2% dalam klasifikasi *ransomware*. Juga dijelaskan bahwa model telah mampu mengidentifikasi ransomware baru secara tepat waktu dan akurat. Namun, penelitian ini menggunakan dataset emulasi. Selain itu, model ini hanya berfungsi dalam mengidentifikasi ransomware, bukan jenis serangan berbasis jaringan lainnya seperti serangan DoS.

5.2. Deep Learning dan Teknologi Big Data

Subbagian ini membahas teknik canggih yang digunakan untuk *deep learning* dan besar teknologi data. Dengan sejumlah besar data yang dihasilkan oleh berbagai industri, prospek dengan minat dalam mengembangkan alat *big data* untuk analisis. Dengan demikian, penulis penelitian [128] miliki mengusulkan kerangka kerja yang menggabungkan Apache Spark dan *Multi-Layer Perceptron* (MLP) menggunakan pembelajaran kaskade. Kerangka kerja terdiri dari tiga tahap, tahap pertama adalah input dataset ke dalam Apache Spark, tahap kedua adalah metode pembelajaran kaskade, dan yang ketiga algoritma deep stage learning diterapkan. Kerangka kerja tersebut telah dievaluasi berdasarkan dua metrik, skor f1 dan akurasi.

Telah mengklaim bahwa dapat memperoleh model yang melakukan analisis *big data* skala besar dalam waktu singkat, dengan lebih sedikit kompleksitas komputasi dan dengan akurasi yang lebih tinggi secara signifikan. Tak perlu dikatakan, keakuratannya dan skor f1 dari model yang diusulkan tidak mencapai bahkan 75% untuk semua tahap. Selanjutnya, teknologi big data terbatas telah dimasukkan ke dalam kerangka kerja yang diusulkan. Selain itu, dalam penelitian [82] penulis telah sangat mengklaim bahwa teknik pembelajaran mesin tidak cukup kuat untuk mendeteksi serangan canggih di IDS yang ada. Karena itu, mereka punya mengusulkan pendekatan terdistribusi untuk deteksi perilaku abnormal dalam jaringan skala besar. Telah menggunakan DBN, ensemble multi-layer SVM, dan Apache Spark untuk mencapainya model.

Model mereka telah dievaluasi menggunakan Area di bawah Karakteristik Operasi Penerima (ROC), ketepatan, daya ingat, pengukuran-f dan waktu pelatihan. Model telah menunjukkan tinggi kinerja dalam mendeteksi perilaku abnormal secara terdistribusi. Selanjutnya, model ini membahas langkah rekayasa fitur untuk pembelajaran ensemble, terutama dengan kumpulan *big data*. Namun, waktu pelatihan untuk pendekatan yang diusulkan mereka jauh lebih tinggi dibandingkan ke model lain yang telah mereka evaluasi. Selanjutnya, jumlah fitur dalam dataset membuat dampak pada keakuratan model.

Selain itu, penulis dalam penelitian [129] telah merancang dan menerapkan kerangka kerja yang melatih DNN menggunakan Apache Spark. Pelatihan model *deep learning* membutuhkan yang besar jumlah data dan luas komputasi. Mereka telah mengklaim bahwa Kerangka kerja dapat mempercepat waktu pelatihan dengan mendistribusikan replika model, melalui keturunan gradien stokastik, di antara node untuk data dalam Sistem File Terdistribusi Hadoop (HDFS). Kerangka kerja tersebut dievaluasi berdasarkan run time, akurasi, dan tingkat kesalahan. Itu Kerangka yang diusulkan telah menunjukkan kinerja waktu dan akurasi yang memuaskan. Sebaliknya, waktu menjalankan model menunjukkan peningkatan ketika jumlah node lebih sedikit. Bahkan, terlihat

bahwa tingkat kesalahan berkurang hanya ketika jumlah iterasi meningkat. Selanjutnya, penelitian [78] telah mengusulkan kerangka kerja untuk melakukan deteksi dan analisis menggunakan RNN dan aturan asosiasi penambangan.

Kerangka kerja ini menggunakan Apache Spark untuk pelatihan setelah dataset dinormalisasi. Kerangka kerja telah dievaluasi menggunakan jumlah dari instance yang diklasifikasikan dengan benar, instance yang salah klasifikasi, Kappa, berarti kesalahan absolut, root kuadrat kesalahan, kesalahan absolut relatif, dan akar kesalahan kuadrat relatif. Pembelajaran mampu mencapai 199 instance dengan klasifikasi benar (100%) dan 0 klasifikasi salah contoh (0%). Selanjutnya, skor Kappa 1 telah tercapai. Di sisi lain, studi telah membatasi model untuk deteksi penyusup saja. Selain itu, metrik evaluasi lainnya belum dipertimbangkan untuk model, seperti waktu pelatihan, ketepatan, dan daya ingat.

Netflow, protokol yang digunakan untuk analisis audit jaringan, dan pemantauan dapat menjadi sumber informasi untuk deteksi insiden dan keperluan forensik. Oleh karena itu, penelitian [109] telah diusulkan sebuah metode yang menggabungkan NetFlows dengan classifier Extreme Learning Machine (ELM), dilatih dalam lingkungan terdistribusi dari Apache Spark untuk deteksi aktivitas malware. Itu Model telah dievaluasi berdasarkan TPR, FPR, presisi, akurasi, tingkat kesalahan, dan f-ukuran. Model yang diusulkan menghasilkan akurasi yang lebih tinggi, lebih sedikit tingkat kesalahan, dan sebagian besar ukuran-f dari skenario. Namun, dalam skenario tertentu metode ini dijalankan dengan akurasi dianggap sebagai tertinggi kedua dibandingkan dengan model lain yang dievaluasi. Selanjutnya, beragam besar teknologi data belum dipertimbangkan. Selain itu, penulis penelitian [110] telah mengusulkan metode deteksi DDoS itu menggunakan jaringan saraf, diimplementasikan pada kluster Apache Spark.

Dengan menerapkan Hadoop HDFS karena kemampuannya menciptakan aplikasi yang toleran terhadap kesalahan dan efisiensi dalam penanganan dataset besar, dikombinasikan dengan jaringan saraf, mereka mampu mencapai akurasi 94%. Mereka telah menegaskan bahwa sistem mereka mampu menangani kecepatan tinggi, dan aliran jaringan volume tinggi secara real-time dan mampu membedakan antara yang asli dan data serangan. Lebih lanjut, mereka mengklaim bahwa Apache Spark cocok untuk pemrosesan yang besar lalu lintas jaringan volume. Kendati demikian, keakuratannya bisa lebih dipupuk menggunakan berbeda algoritma *deep learning* atau dengan memasukkan metode optimasi. Juga, modelnya saja berlaku untuk mendeteksi tipe serangan tunggal. Selain itu, dalam penelitian [78] penulis telah mengusulkan sistem yang menggabungkan dua pendekatan, yaitu ANN terdistribusi berbasis anomali, dan pendekatan berbasis tanda tangan.

Untuk detektor berbasis anomali, BigDL deep learning library digunakan di atas Apache Spark. Untuk pendekatan berbasis tanda tangan, Suricata IDS open source digunakan. Model mereka miliki telah dievaluasi berdasarkan FPR, akurasi, dan DR. Model hibrida mereka telah mengungguli detektor berbasis tanda tangan tradisional, dan detektor anomali berbasis saraf. Di sisi lain, metrik terbatas telah digunakan untuk mengevaluasi model. Demikian juga, modelnya hanya terbatas untuk mendeteksi satu jenis serangan. Selanjutnya, penulis penelitian [81] telah mengusulkan metode deteksi anomali yang digunakan RBM dan RNN untuk deteksi anomali di jaringan listrik. Penulis terutama menggunakan Apache Hadoop dan Apache Spark untuk menganalisis sumber data heterogen dalam kekuasaan *big data*, dan untuk menerapkan kerangka *deep learning*. Model mereka telah dievaluasi berdasarkan pada akurasi, FPR, dan FNR.

Mampu mencapai tingkat akurasi tinggi, FPR rendah, dan FNR rendah. Namun, model ini belum dilatih tentang dataset patokan. Sebagai tambahan, metrik evaluasi terbatas telah digunakan untuk evaluasi model. Selain itu, dalam penelitian [38] penulis telah membahas kerangka kerja intrusi real-time deteksi. Mereka telah menggunakan jaringan saraf CC4 yang diusulkan dalam penelitian [130] dan MLP. Lebih lanjut, mereka telah menggunakan Apache Storm untuk mengalirkan data untuk pemrosesan real-time. Telah menegaskan bahwa waktu pelatihan melihat pengurangan yang signifikan ketika menggunakan Apache Badai dibandingkan dengan metode biasa. Mereka telah mengevaluasi model berdasarkan akurasi, FPR, waktu pelatihan, dan FNR. Mereka telah mencapai akurasi 89% dan FPR 4,32%. Namun, akurasi rata-rata turun di bawah 90%, yang dapat lebih ditingkatkan dengan memasukkan algoritma *deep learning* lainnya. Selain itu, percobaan hanya dilakukan pada satu bidang dataset, keterbatasan, dari studi yang telah memasukkan *deep learning* dan teknologi *big data*.

5.3. Deep Learning dan Teknologi Big Data untuk Keamanan IoT

Subbagian Sub bagian ini membahas hubungan antara tiga bidang utama penelitian ini. Selanjutnya, kami telah menguraikan teknik-teknik canggih untuk *deep learning*, *big data* teknologi dan keamanan IoT. Selain itu, kami telah mentabulasikan kombinasi penelitian digunakan di negara-of-the-art dan mengidentifikasi penggunaan *deep learning*,

teknologi *big data*, dan Keamanan IoT dalam setiap studi ini. Akhirnya, kami telah membahas beberapa yang digunakan secara mencolok infrastruktur cloud yang mendukung *deep learning*, teknologi *big data*, dan keamanan IoT. Berdasarkan analisis kritis kami, kami telah berupaya mengatasi hubungan di antara keduanya *deep learning*, *big data*, dan keamanan IoT. Namun, studi sebelumnya hanya dimasukkan baik *deep learning* dan keamanan IoT atau *deep learning* dan teknologi *big data*.

Selanjutnya, studi minimal telah dilakukan pada *deep learning*, teknologi big data, dan IoT keamanan. Ini jelas membuatnya jelas bahwa ada area khusus untuk peneliti masa depan alamat. Selain itu, dengan upaya maksimal kami untuk menganalisis berbagai studi secara kritis, kami telah mampu mengidentifikasi hanya dua studi yang telah membahas ketiga komponen tersebut. Kelebihan dan kekurangan dari kedua studi telah dijelaskan di bawah ini. Karena pertumbuhan eksponensial dari berbagai perangkat yang saling berhubungan, serangan inovatif terjadi sedang dilakukan pada perangkat ini. Karena itu, perlu untuk datang dengan inovatif dan metodologi bukti untuk mencegah insiden / bencana. Oleh karena itu, penulis [29] memiliki merancang kerangka *big data* untuk deteksi intrusi menggunakan metode klasifikasi seperti, DNN, SVM, *random tree*, pohon keputusan, dan Bayes naif. Metrik yang digunakan untuk evaluasi adalah akurasi, daya ingat, tingkat false, spesifisitas, dan waktu prediksi.

Apache Spark telah digunakan sebagai platform untuk menerapkan deteksi intrusi di smart grid menggunakan analitik *big data*. Mereka mengklaim bahwa algoritma DNN mendapatkan akurasi tertinggi untuk dataset mentah. Meskipun demikian, akurasi tertinggi yang diperoleh adalah oleh model DNN, tetapi akurasinya kurang dari 80%. Selain itu, waktu prediksi DNN lebih tinggi dibandingkan dengan model lain. Selain itu, penulis dalam penelitian ini [30] telah membahas kemajuan dalam perangkat keras, perangkat lunak, dan topologi jaringan, termasuk IoT, menimbulkan ancaman keamanan yang memerlukan modern pendekatan hari yang akan diterapkan. Dengan demikian, mereka telah mengusulkan IDS berbasis DNN. Itu DNN yang digunakan adalah MLP bersama dengan FFNN. Telah dibahas bahwa kerangka kerjanya telah dikembangkan berdasarkan teknologi big data, platform komputasi cluster Apache Spark. Itu Komputasi cluster spark Apache adalah setup melalui Apache Hadoop Yet Another Resource Negosiator. Telah mengevaluasi model berdasarkan akurasi, presisi, daya ingat, f-score, TPR, dan FPR. Selain itu, model mereka telah mengungguli semua mesin tradisional lainnya pendekatan pembelajaran di HIDS dan NIDS. Namun, dalam kalsifikasi multi-kelas akurasi turun di bawah 90% untuk serangan tertentu di beberapa dataset. Selanjutnya, DNN tidak dilatih tentang dataset patokan IDS. Tabel 6. membandingkan studi berdasarkan inklusi *deep learning* teknologi *big data*, dan keamanan IoT.

Tabel 6. Pembelajaran Jauh, Teknologi Big Data dan Keamanan IoT

Studi	Deep Learning	Teknologi Big Data	Keamanan IoT
[39]	✓	✓	
[129]	✓	✓	
[78]	✓	✓	
[131]	✓	✓	
[128]	✓	✓	
[81]	✓	✓	
[17]	✓		✓
[18]	✓		✓
[19]	✓		✓
[85]	✓		✓
[132]	✓		✓
[40]	✓		✓
[127]	✓		✓
[82]	✓	✓	
[133]	✓	✓	
[77]	✓	✓	
[89]	✓		✓
[86]	✓		✓
[87]	✓		✓

Seperti yang terlihat dari Tabel 6, hanya *deep learning* dan keamanan IoT atau *deep learning* dan *big data* teknologi telah dimasukkan dalam studi ini. Keberhasilan implementasi studi [23] dan [24], meyakinkan para peneliti bahwa *deep learning* dan teknologi *big data* dapat digabungkan untuk keamanan IoT. Karena itu, karena terbatasnya penelitian yang dilakukan pada bidang-bidang ini, kami menganjurkan beberapa peneliti pada masa depan untuk mengimplementasikan model berdasarkan berbagai algoritma *deep learning*, dan teknologi *big data* untuk keamanan IoT.

5.4. Infrastruktur Cloud untuk Deep Learning, Teknologi Big Data dan Keamanan IoT

Subbagian ini merinci infrastruktur cloud yang dapat diterapkan untuk *deep learning*, teknologi *big data*, dan keamanan IoT. *Deep learning* telah menunjukkan hasil yang menjanjikan di banyak domain, namun *deep*

learning mungkin cukup komputasional dalam skala besar aplikasi. Ini pada gilirannya, memaksa masuknya sumber daya komputasi tambahan. Kapan *deep learning* diterapkan pada aplikasi skala besar, sumber daya yang ada mungkin terbatas. Oleh karena itu, infrastruktur cloud dapat digunakan untuk mengatasi tantangan ini karena mengandung banyak jumlah sumber daya seperti, CPU multi-core, GPU multi-core, memori, dan bandwidth. Selain itu, beberapa infrastruktur cloud bahkan menyediakan dukungan untuk teknologi *big data* dan IoT. Selanjutnya mentabulasi beberapa layanan cloud populer dan dukungan mereka untuk *deep learning*, teknologi big data dan IoT pada Tabel 7.

Tabel 7. Infrastruktur Awan untuk Deep Learning, Teknologi Big Data dan IoT

Layanan Cloud	Mendukung Deep Learning	Mendukung Teknologi Big Data	Mendukung IoT
Google Cloud	✓	✓	✓
AWS Sagemaker	✓	✓	✓
Deep Cognition	✓	✓	✓
IBM Watson	✓	✓	✓
Microsoft Azure	✓	✓	✓
Oracle Cloud	✓	✓	✓
Alibaba Cloud	✓	✓	✓
Tensor Pad	✓	-	-

Kemungkinan cloud yang berkembang telah berkontribusi pada pertumbuhan Crimeware-asa-Service (CaaS), yang memungkinkan penjahat cyber dengan keahlian teknis terbatas untuk melakukan serangan terorganisir dan otomatis [134]. Ada banyak jenis layanan yang disediakan oleh CaaS seperti, layanan broker bayangan, kit eksploitasi Neutrino, perangkat Mirai untuk disewakan, DiamondFox layanan malware modular, Tox ransomware-as-a-service, dan phishing-as-a-service.

VI. TANTANGAN TERBUKA DAN ARAH DI MASA DEPAN

Bagian ini menyoroti tantangan penelitian yang paling signifikan dalam hal keamanan IoT menggunakan *deep learning* dan teknologi *big data*. Kemampuan canggih di IoT keamanan, *deep learning*, dan teknologi *big data* telah diperiksa untuk menentukan tantangan penelitian utama, saran, dan arah masa depan.

6.1. Deteksi Ancaman Kemanan

Karena kecepatan tinggi dan variasi dalam beberapa aplikasi IoT domain, struktur yang kompleks data membuatnya lebih menantang untuk mendeteksi ancaman keamanan. Selanjutnya, memilih serangkaian fitur yang diakui untuk analitik keamanan dalam algoritme pembelajaran dalam dapat menarik [135]. Mekanisme yang ada kurang efisien dalam menemukan korelasi tersembunyi di antara fitur ini. Lebih lanjut, algoritma *deep learning* yang muncul dapat menangani parameter tersembunyi dari aplikasi IOT. Selain itu, *deep learning* mampu menemukan korelasinya dalam berbagai data. Selain itu, dimungkinkan untuk memperoleh tingkat deteksi tinggi untuk dideteksi serangan zero-day lebih efisien [136]. Terakhir, dibandingkan dengan pendekatan tradisional, distribusi representasi dari algoritma *deep learning* dapat menangani pemilihan banyak fitur dengan data yang luar biasa untuk mengekstrak informasi untuk aplikasi IoT multi-domain [137].

6.2. Durasi Pelatihan

Teknik yang ada membutuhkan waktu lebih lama untuk melatih model untuk deteksi yang akurat. Demikian juga karena, mereka membutuhkan dataset besar untuk melatih model [133]. Kedua kondisi ini adalah utama hambatan dalam mekanisme saat ini, namun kemampuan algoritma *deep learning* untuk gunakan lebih sedikit durasi pelatihan dan set data memungkinkan untuk menangani model secara efisien. Sebagai tambahan, ukuran bets juga dapat memengaruhi waktu yang digunakan untuk pelatihan karena akumulasi jaringan setelah pembaruan berat [85].

Tantangan-tantangan ini harus ditangani oleh opsi pelapisan ganda dalam *deep learning* yang membantu menimbang dan mengenali rangkaian spesifik parameter dari *dataset*. Terakhir, fasilitas pemrosesan dan penyimpanan terbatas lebih lanjut menghambat waktu pelatihan model. Sebaliknya, teknologi *big data* dan *cloud based* arsitektur harus meningkatkan kemanjuran model dengan membatasi durasi pelatihan [133].

6.3. Kompleksitas Waktu

Sebagian besar teknik deteksi yang ada telah dikembangkan untuk aplikasi pemrosesan batch dan bukan untuk deteksi waktu nyata. Kompleksitas waktu berperan penting dalam pendeteksian ancaman dalam aplikasi IoT, yang berisi lebih banyak data streaming. Selanjutnya, ini membantu untuk mengidentifikasi dampak dari beberapa atribut yang terlibat dalam ancaman keamanan. Studi lain telah disorot bahwa terlepas dari menggunakan data waktu nyata besar-besaran pendekatan yang ada paling umum adalah tidak efektif dalam mengklasifikasikan intrusi karena mereka menggunakan pembelajaran yang dangkal [19].

Apalagi ini masalah kompleksitas waktu dapat diselesaikan dengan mudah dalam pendekatan *deep learning* dengan menerapkan Komponen GPU, karena membantu dalam pemrosesan waktu nyata dan sangat efisien dalam analisis ancaman secara *real-time* [34]. Selanjutnya, mempekerjakan Apache Spark atau Apache Hadoop efektif dalam meminimalkan kompleksitas waktu [128].

6.4. Komputasi dalam Memori

Pemrosesan dalam memori adalah teknologi pengembangan yang sedang tren untuk memproses data disimpan dalam basis data dalam memori. Ini memainkan peran penting dalam analitik streaming dan memorycentric Arsitektur. Teknik konvensional didasarkan pada penyimpanan disk dan relasional database yang menghadapi banyak tantangan untuk menangani volume data modern dari perangkat IoT. Selanjutnya, teknik-teknik ini menjadi tidak memadai untuk diintegrasikan dengan analisis keamanan yang dihasilkannya organisasi menjadi lebih rentan dalam hal keamanan. Dalam database relasional, data disimpan dalam beberapa tabel dan perlu menggunakan SQL untuk melakukan pemrosesan permintaan. Ini ada pendekatan lebih lanjut menimbulkan kesulitan dalam menggabungkan dan mengumpulkan data untuk diproses dan SQL dirancang untuk mengambil baris data sebelum diproses.

Masalah yang disebutkan di atas akan menjadi mudah ditangani oleh pemrosesan dalam memori untuk analitik keamanan. Data yang disimpan dengan cepat diakses ketika disimpan dalam RAM atau memori flash dibandingkan dengan penyimpanan disk. Selanjutnya, memori pemrosesan memungkinkan data dianalisis secara waktu nyata. Pemrosesan real-time membantu membuat pelaporan dan pengambilan keputusan yang lebih cepat untuk ancaman keamanan. Teknologi *big data* modern seperti Apache Spark dan Apache Flink memproses data mereka dalam memori. Menggabungkan ini teknologi untuk mengembangkan analitik keamanan baru akan meningkatkan kinerja dan efisiensi untuk analitik keamanan [3] [138] [139].

6.5. Keterbatasan Komputasi dalam Energi

Kompleksitas komputasi adalah salah satu tantangan terpenting dalam bidang IoT keamanan perangkat, *deep learning*, dan *big data*, bidang penelitian. Perangkat IoT dioperasikan di baterai berdaya rendah dan CPU-nya memiliki tingkat clock yang lebih rendah. Melakukan perhitungan apa pun dalam perangkat IoT harus cepat dan harus meminimalkan operasi langsung [58]. Sebagai gantinya, komputasi harus dilakukan dalam cloud atau komputasi tepi. Demikian pula, studi telah menyoroti bahwa menerapkan sistem keamanan berbasis algoritmik harus focus lebih lanjut tentang memproduksi sistem perhitungan ringan untuk analisis [140]. Di samping itu, pertumbuhan big data serta peningkatan daya komputasi menguntungkan *deep learning* teknik untuk tumbuh dengan cepat, yang pada gilirannya telah digunakan dalam industri serval [19]. Lebih lanjut, perhitungan dapat dioptimalkan menggunakan sifat-sifat komputasi terdistribusi dan terdistribusi algoritma.

Operasi algoritma ini dilakukan di jaringan hibrida, di mana pekerjaan didistribusikan ke berbagai mesin untuk meningkatkan efisiensinya [24]. Beberapa tantangan yang dibahas di atas telah dengan mudah ditangani oleh Apache Spark streaming besar kerangka kerja teknologi data, yang mampu memanfaatkan RDD, Dstreams dan parallel fitur komputasi untuk memproses data dengan perhitungan yang layak [128].

6.6. Sisi Keamanan

Platform komputasi memungkinkan skalabilitas yang lebih besar untuk proses komputasi dan penyimpanan kekuatan untuk perangkat IoT. Selanjutnya, akan memberikan peluang ke perangkat yang berada di dekat ke sumber data, yang memungkinkan operasi cerdas dilakukan jauh dari terpusat titik infrastruktur. Sementara itu, infrastruktur cloud edge dalam jaringan tetap ada sumber data IoT, terutama yang berkaitan dengan komputasi jaringan untuk menyediakan yang cerdas layanan tepi untuk mendeteksi ancaman apa pun secara real-time. Apalagi perangkat IoT tidak memiliki cukup sumber daya untuk menyimpan dan menganalisis data untuk ancaman apa pun [133]. Dengan demikian, mengadopsi komputasi tepi akan memfasilitasi untuk menangani tantangan di atas dengan mendistribusikan proses ke berbagai sumber *over cloud* untuk analisis [141]. Terakhir, mengintegrasikan *deep learning* dan teknologi *big data* untuk analisis keamanan perangkat IoT menyediakan sistem pemrosesan yang lebih efisien untuk dan secara akurat mendeteksi ancaman

VII. KESIMPULAN

Populasi yang berkembang dari perangkat IoT telah berkontribusi pada pertimbangan keamanan risiko yang terkait dengan mereka. Perangkat IOT terbukti rentan karena baru-baru ini meningkatnya serangan seperti, botna Carna dan Mirai. Selain itu, perangkat IoT menghasilkan volume besar, kecepatan dan variasi data. Ini membuat solusi yang ada kurang efisien dan

membutuhkan solusi modern. Dalam hal ini, *deep learning* telah diterima secara luas di kalangan peneliti dan organisasi karena akurasi tinggi, kemampuan untuk mempelajari fitur mendalam, dan pengawasan manusia yang minimal. Selain itu, teknologi big data juga menarik karena kemampuan mereka dalam memproses sejumlah besar data, bersama dengan kemampuan mereka untuk memproses data dalam berbagai lingkungan seperti waktu nyata, kumpulan, dan aliran.

Karenanya, studi ini telah menyelidiki kemungkinan memasukkan teknologi *deep learning* dan *big data* untuk keamanan IoT. Temuan kami menunjukkan bahwa banyak penelitian telah tergabung dalam belajar dengan keamanan IoT atau *deep learning* dengan teknologi *big data*, namun, ada kurangnya penelitian dalam menggabungkan teknologi *deep learning* dan *big data* untuk keamanan IoT, namun demikian, penyelidikan kami telah mengungkapkan bahwa dua penelitian telah membuktikan efisiensi dan kelayakan menggabungkan *deep learning* dan teknologi *big data* untuk keamanan IoT berakhir model tradisional. Mempertimbangkan berbagai persyaratan keamanan IoT yang dibahas dan tantangan yang dibahas telah merencanakan untuk mengusulkan sebuah novel. kerangka kerja untuk keamanan IoT berdasarkan *deep learning* dan teknologi *big data* dan kinerja analisis eksperimental untuk membuktikan kemanjurannya, dalam waktu dekat.

Selanjutnya, berusaha memecahkan masalah yang dibahas dalam menggabungkan teknologi *deep learning* dan *big data* untuk keamanan IoT.

DAFTAR PUSTAKA

[1] L. Deng, “Three Classes of Deep Learning Architectures and Their Applications: A Tutorial Survey,” in *APSIPA Transactions on Signal and Information Processing*, 2012.

[2] N. Mohan, “Edge-fog cloud: A distributed cloud for internet of things computations,” in *Proceeding Cloudification of the Internet of Things (CIoT)*, 2016, pp. 1–6.

[3] R. A. A. Habeeb, “Real-time big data processing for anomaly detection,” *Int. J. Inf. Manage.*, vol. 45, pp. 289–307, 2019.

[4] D. G, “Trending: Iot malware attacks of 2018,” 2018. [Online]. Available: https://securingtomorrow.mcafee.com/consumer/mobile-and-iot-security/top-trend%0Aing-iot-malware-attacks-of-2018/.

[5] W. G.Wong, “Developers Discuss IoT Security and Platforms Trends,” 2019. [Online]. Available: https://www.electronicdesign.com/embedded/developers-discuss-iot-security-and-p%0Alatforms-trends.

[6] K. Lab, “New trends in the world of iot threats,” 2019. [Online]. Available: https://securelist.com/new-trends-in-the-world-of-iot-threats/87991/.

[7] K. A, “Issues, Challenges, Tools and Good Practices,” in *2013 Sixth International Conference on Contemporary Computing (IC3)*, 2013.

[8] A. A. Cardenas, “Big Data Analytics for Security,” *IEEE Secur. Priv.*, vol. 11, no. 6, pp. 74–76, 2013.

[9] C. D. McDermott, “Botnet Detection in The Internet of Things using Deep Learning Approaches,” in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8.

[10] M. Aly, “Enforcing Security in Internet of Things Frameworks: A Systematic Literature Review,” in *Internet of Things*, 2019.

[11] J. Pan, “Cybersecurity Challenges and Opportunities in The New Edge Computing +IoT World,” in *Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, 2018, pp. 29–32.

[12] A. F. A. Rahman, “Securing Sensor to Cloud Ecosystem using Internet of Things (IoT) Security Framework,” in *Proceedings of the International Conference on Internet of things and Cloud Computing, ACM*, 2016, p. 79.

[13] C. Perera, “Big Data Privacy in The Internet of Things Era,” *IT Prof.*, vol. 17, no. 3, pp. 32–39, 2015.

[14] B. Kolosnjaji, “Deep learning for classification of malware system call sequences,” in *Australasian Joint Conference on Artificial Intelligence, Springer*, 2016, pp. 137–149.

[15] Z. Yuan, “Droid-Sec: Deep Learning in Android Malware Detection,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 371–372, 2014.

[16] Z. Yuan, “Droiddetector: Android Malware Characterization and Detection using Deep Learning,” *Tsinghua Sci. Technol.*, vol. 21, no. 1, 2016.

[17] Y. Meidan, “Nbaiot—Network-Based Detection of IoT Botnet Attacks using Deep Autoencoders,” *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 12–22, 2018.

[18] A. A. Diro, “Distributed Attack Detection Scheme using Deep Learning Approach for Internet of tThings,” in *Future Generation Computer Systems*, 2018, pp. 761–768.

[19] B. Roy, “A Deep Learning Approach for Intrusion Detection in Internet of Things using Bi-directional Long Short-Term Memory Recurrent Neural Network,” in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, 2018, pp. 1–6.

[20] R. A. A. Habeeb, “Clustering-Based Real-

Time Anomaly Detection a Breakthrough in Big Data Technologies," in *Transactions on Emerging Telecommunications Technologies*, 2018.

[21] G. P. Gupta, "A Framework for Fast and Efficient Cyber Security Network Iintrusion Detection using Apache Spark," in *Procedia Computer Science*, 2016, pp. 824–831.

[22] V. P. Janeja, "B-dids: Mining Anomalies in a Big-Distributed Intrusion Detection System," in *2014 IEEE International Conference on Big Data (Big Data)*, 2014, pp. 32–34.

[23] K. Vimalkumar, "A Big Data Framework for Intrusion Detection in Smart Grids using Apache Spark," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 198–204.

[24] R. Vinayakumar, "Deep Learning approach for Intelligent Intrusion Detection System," in *IEEE Access*, 2019, pp. 41525–41550.

[25] C. Cimpanu, "Sirenjack attack lets hackers take control over emergency alert sirens," 2019. [Online]. Available: https://www.bleepingcomputer.com/news/security/sirenjack-attack-lets-hackers-t%0Aake-control-over-emergency-alert-sirens/.

[26] J. Sanders, "5 biggest iot security failures of 2018," 2019. [Online]. Available: https://www.techrepublic.com/article/5-biggest-iot-security-failures-of-2018/.

[27] L. Mathews, "Hackers use ddos attack to cut heat to apartments," 2016. [Online]. Available: https://www.forbes.com/sites/leemathews/2016/11/07/ddos-attack-leaves-finnish-a%0Apartments-without-heat/#4bd0d0961a09.

[28] "Iot Role in Dyn Cyberattack." [Online]. Available: https://www.kaspersky.com/blog/attack-on-dyn-explained/13325/.

[29] D. Etherington, "Large ddos attacks cause outages at twitter, spotify, and other sites – techcrunch," 2016. [Online]. Available: https://techcrunch.com/2016/10/21/many-sites-including-twitter-and-spotify-suf%0Afering-outage/.

[30] "The Possible Vendetta Behind The East Coast Web Slowdown," 2019. [Online]. Available: https://www.bloomberg.com/news/articles/2016-10-21/internet-service-disrupted-i%0An-large-parts-of-eastern-u-s.

[31] R. A. A. Habeeb, "Clustering-based real-time anomaly detection—a breakthrough in big data technologies," in *Transactions on Emerging Telecommunications Technologies*, 2018.

[32] A. Schiffer, "How a Fish Tank Helped Hack a Casino," 2107. [Online]. Available: https://www.washingtonpost.com/news/innovations/wp/2017/07/21/how-a-fish-tank-h%0Aelped-hack-a-casino/?utm_term=.8ba4c46540ef.

[33] T. Dodrill, "Hacker Turns Baby Monitor Into Real Life Nightmare," 2019. [Online]. Available: https://www.offthegridnews.com/privacy/hacker-turns-baby-monitor-into-real-lif%0Ae-nightmare/.

[34] Y. Guo, "Deep Learning for Visual Understanding: A Review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[35] N. Papernot, "The Limitations of Deep Learning in Adversarial Settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016, pp. 372–387.

[36] R. Shokri, "Privacy-Preserving Deep Learning," in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1310–1321.

[37] J. Wang, "Deep Learning for Sensor-Based Activity Recognition," in *Pattern Recognition Letters*, 2019, pp. 3–11.

[38] M. Strohbach, "Towards a Big Data Analytics Framework for IoT and Smart City Applications," in *in: Modeling and Processing for Next-Generation Big-Data Technologies*, 2015, pp. 257–282.

[39] G. Mylavarapu, "Real-time Hybrid Intrusion Detection System using Apache Storm," in *High Performance Computing and Communications IEEE 7th Int. Symp. Cyberspace Safety and Security Conf. Embedded Software and Systems*, 2015, pp. 1436–1441.

[40] Y. Zhou, "Deep Learning Approach for Cyber attack Detection," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2018, pp. 262–267.

[41] C. Wang, "Deep Learning Semisupervised Salient Object Detection in The Fog of IoT," in *IEEE Transactions on Industrial Informatics*.

[42] M. Mohammadi, "Deep Learning for IoT Big Data and Streaming Analytics: A Survey," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018.

[43] A. Gandomi, "Beyond the Hype: Big Data Concepts, Methods, and Analytics," *Int. J. Inf. Manage.*, vol. 35, no. 2, pp. 137–144, 2015.

[44] K. Adam, "Bigdata: Issues, Challenges, Technologies and Methods," in *Proceedings of the International Conference on Data Engineering 2015 (DaEng-2015)*, 2019, pp. 541–550.

[45] V. K. Vavilapalli, "Apache Hadoop Yarn: Yet Another Resource Negotiator," in *Proceedings of the 4th annual Symposium*

on Cloud Computing, 2013, p. 5.

[46] M. Zaharia, "Apache Spark: A Unified Engine for Big Data Processing," *Commun. ACM*, vol. 59, no. 11, pp. 56–65, 2016.

[47] J. S. van der Veen, "Dynamically Scaling Apache Storm for The Analysis of Streaming Data," in *2015 IEEE First International Conference on Big Data Computing Service and Applications*, 2015, pp. 154–161.

[48] P. Carbone, "Apache flink: Stream and Batch Processing in a Single Engine," *Bull. IEEE Comput. Soc. Tech. Comm. Data Eng.*, vol. 36, no. 4, 2018.

[49] A. Chebotko, "A Big Data Modeling Methodology for Apache Cassandra," in *IEEE International Congress on Big Data*, 2015, pp. 238–245.

[50] D. Borthakur, "Apache Hadoop Goes Realtime at fFacebook," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, 2011, pp. 1071–1080.

[51] M. Marjani, "Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges," in *IEEE Access*, 2017, pp. 5247–5261.

[52] S. Moin, "Securing IoTs in Distributed Blockchain: Analysis, Requirements and Open Issues," in *Future Generation Computer Systems*, 2019, pp. 325–343.

[53] F. X. Ming, "Real-time Carbon Dioxide Monitoring Based on IoT & Cloud Technologies," in *Proceedings of the 2019 8th International Conference on Software and Computer Applications*, 2019, pp. 517–521.

[54] M. A. Khan, "Iot security: Review, Blockchain Solutions, and Open Challenges," in *Future Generation Computer Systems*, 2018, pp. 395–411.

[55] J.-S. Cho, "Securing Against Brute-force Attack: A Hash-Based rfid Mutual Authentication Protocol using a Secret Value," *Comput. Commun.*, vol. 34, no. 3, pp. 391–397, 2011.

[56] H. A. Khattak, "Perception Layer Security in Internet of Things," in *Future Generation Computer Systems*, 2019, pp. 144–164.

[57] "Spark Security," 2019. [Online]. Available: https://spark.apache.org/docs/latest/security.html.

[58] M. M. Hossain, "Towards an Analysis of Security Issues, Challenges, and Open Problems in The Internet of Things," in *015 IEEE World Congress on Services*, 2015, pp. 21–28.

[59] "How-to: Do Data Quality Checks using Apache Spark Dataframes," 2019. [Online]. Available: https://blog.cloudera.com/blog/2015/07/how-to-do-data-quality-checks-using-apa%0Ache-spark-dataframes/.

[60] S. Babar, "Proposed Security Model and Threat Taxonomy for the Internet of Things (IoT)," in *International Conference on Network Security and Applications*, 2010, pp. 420–429.

[61] "Spark Standalone Mode," 2019. [Online]. Available: https://spark.apache.org/docs/latest/spark-standalone.html#high-availability.

[62] J. H. Ziegeldoorf, "Privacy in The Internet of Things: Threats and Challenges," *J. Comput. Mach.*, vol. 7, no. 1, pp. 110–119, 2018.

[63] Y. Bengio, "Learning Deep Architectures for AI," *Found. trends® Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[64] L. Deng, "Deep Learning: Methods and Aapplications," *Found. Trends® Signal Process.*, vol. 7, no. 3–4, pp. 197–387, 2014.

[65] G. Huang, "Semi-Supervised and Unsupervised Extreme Learning Machines," *IEEE Trans. Cybern.*, vol. 4, no. 12, pp. 2405–2417, 2014.

[66] C.-Y. Liou, "Modeling Word Perception using The Elman Network," in *Neurocomputing*, 2008, pp. 3150–3157.

[67] C.-Y. Liou, "Autoencoder for Words," in *Neurocomputing*, 2014, pp. 84–96.

[68] T. A. Tang, "Deep Recurrent Neural Network for Intrusion Detection in Sdn-Based Networks," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, 2018, pp. 202–206.

[69] B. Li, "Using Stochastic Computing to Reduce The Hardware Requirements for a Restricted Boltzmann Machine Classifier," in *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2016, pp. 36–41.

[70] A. M. Abdel-Zaher, "Breast Cancer Classification Using Deep Belief Networks," in *Expert Systems with Applications*, 2016, pp. 139–144.

[71] G. E. Hinton, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[72] G. Hinton, "Deep Belief Networks," *Scholarpedia*, vol. 4, no. 5, p. 5947, 2009.

[73] H. Sak, "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[74] Y. LeCun, "Handwritten Digit Recognition with a Back-propagation Network," in *Advances in neural information processing systems*, 1990, pp. 396–404.

[75] Y. LeCun, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[76] M. Liang, "Recurrent Convolutional Neural

Network for Object Recognition," in *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3367–3375.

[77] S. Alzahrani, "Detection of Distributed Denial of Service (Ddos) Attacks Using Artificial Intelligence on Cloud," in *2018 IEEE World Congress on Services (SERVICES)*, 2018, pp. 35–36.

[78] A. Thilina, "Intruder Detection using Deep Learning and Association Rule Mining," in *2016 IEEE International Conference on Computer and Information Technology (CIT)*, 2016, pp. 615–620.

[79] A. Elsaeidy, "A Smart City Cyber Security Platform for Narrowband Networks," in *Proceeding 27th International Telecommunication Networks and Applications Conference (ITNAC)*, 2017, pp. 1–6.

[80] A. Dawoud, "Deep learning and software-defined networks : Towards secure iot architecture," in *Internet of Things*, 2018, pp. 82–89.

[81] L. Dong-Lan, "A Multilevel Deep Learning Method for Data Fusion and Anomaly Detection of Power Big Data," in *3rd Annual International Conference on Electronics, Electrical Engineering and Information Science (EEEIS 2017)*, 2017.

[82] N. Marir, "Distributed Abnormal Behavior Detection Approach Based on Deep Belief Network and Ensemble SVM using Spark," in *IEEE Access*, 2018, pp. 59657–59671.

[83] Y. He, "Real-Time Detection of False Data Injection Aattacks in Smart Grid: A Deep Learning-Based Intelligent Mechanism," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2505–2516, 2017.

[84] J. Chauhan, "Breathing-Based Aauthentication on Resource-Constrained IoT Devices Using Recurrent Neural Networks," *Computer (Long. Beach. Calif).*, vol. 51, no. 5, pp. 60–67, 2018.

[85] H. HaddadPajouh, "A Deep Recurrent Neural Network Based Aapproach for Internet of Tthings Malware Threat Hunting," in *Future Generation Computer Systems*, 2018, pp. 88–96.

[86] M. Roopak, "Deep Learning Models for Cyber Security in IoT Networks," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, 2019, pp. 0452–0457.

[87] S. Homayoun, "DRTHIS: Deep Ransomware Threat Hunting and Intelligence System at The Fog layer," in *Future Generation Computer Systems*, 2019, pp. 94–104.

[88] J. Su, "Lightweight Classification of IoT Malware Based on Image Recognition," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, 2018.

[89] A. Azmoodeh, "Robust Malware Detection for Internet of (battlefield) things devices using deep eigenspace learning," *IEEE Trans. Sustain. Comput.*, vol. 4, no. 1, pp. 88–95, 2019.

[90] W. G. Hatcher, "A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends," in *IEEE Access*, 2018, pp. 24411–24432.

[91] H. Ma, "A Theano-based Distributed Training Framework," in *European Conference on Parralel Processing*, 2016, pp. 800–813.

[92] P. Roy, "Numa-caffe: Numa-aware Deep Learning Neural Networks," *ACM Trans. Archit. Code Optim.*, vol. 15, no. 2, p. 24, 2018.

[93] N. Ketkar, "Introduction to Pytorch," in *Deep lLarning with Python*, 2017, pp. 195–208.

[94] M. Ravanelli, "The Pytorch-kaldi Speech Recognition Toolkit," in *Proceeding Speech and Signal Processing (ICASSP) ICASSP 2019 -2019 IEEE Int. Conf. Acoustics*, 2019, pp. 6465–6469.

[95] F. Seide, "CNTK: Microsoft's Open-Source Deep-learning Toolkit," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 2135–2135.

[96] A. Candel, "Deep Learning with H20," in *H20*.

[97] A. Parvat, "A Survey of Deep-learning Frameworks," in *Proceeding International Conference Inventive Systems and Control (ICISC)*, 2017, pp. 1–7.

[98] "What is a Confusion Matrix in Machine Learning," 2019. [Online]. Available: https://machinelearningmastery.com/confusion-matrix-machine-learning/.

[99] J. Han, *Data Mining: Concepts and Techniques*, 3rd Editio. San Francisco: Morgan Kaufmann Publishers Inc, 2011.

[100] D. M. Powers, "Evaluation: From Precision, Recall and F-measure to Roc, Informedness, Markedness and Correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.

[101] M. Sokolova, "A Systematic Analysis of Performance Measures for Classification Tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009.

[102] "What is a false positive rate?," 2019. [Online]. Available: https://www.corvil.com/kb/what-is-a-false-positive-rate.

[103] Y. Xin, "Machine Learning and Deep Learning Methods for Cybersecurity," in *IEEE Access*, 2018, pp. 35365–35381.

[104] "Matthews Correlation Coefficient," 2019. [Online]. Available: https://scikit-

learn.org/stable/modules/generated/sklearn.metrics.matthews%5C_co%0Arrcoef.html.

[105] B. Matthews, “Comparison of The Predicted and Observed Secondary Structure of t4 Phage Lysozyme,” in *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 1975, pp. 442–451.

[106] M. L. McHugh, “Interrater Reliability: The Kappa Statisti,” *Biochem. Medica Biochem. Medica*, vol. 22, no. 3, pp. 276–282, 2012.

[107] M. Ahmed, “A Survey of Network Anomaly Detection Techniques,” *J. Netw. Comput. Appl.*, vol. 60, pp. 19–31, 2016.

[108] M. Nobakht, “A Host-based Intrusion Detection and Mitigation Framework for Smart Home IoT using Openflow,” in *2016 11th International conference on availability, reliability and security (ARES)*, 2016, pp. 147–156.

[109] J. Saxe, “Deep Neural Network Based Malware Detection using Two Dimensional Binary Program Features,” in *2015 10th International Conference on Malicious and Unwanted Software (MALWARE)*, 2015, pp. 11–20.

[110] I. Kara, “Static and Dynamic Analysis of Third Generation Cerber Ransomware,” in *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, 2018, pp. 12–17.

[111] J. M. Ceron, “Improving IoT Botnet Investigation Using an Adaptive Network Layer,” *Sensor*, vol. 19, no. 3, p. 727, 2019.

[112] C. Kolias, “Ddos In The IoT: Mirai and Other Botnets,” *Computer (Long. Beach. Calif).*, vol. 50, no. 7, pp. 80–84, 2017.

[113] S. Vashi, “Internet of Things (IoT): A Vision, Architectural Elements, and Security Issues,” in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, 2017, pp. 492–496.

[114] I. Andrea, “Internet of Things: Security Vulnerabilities and Challenges,” in *2015 IEEE Symposium on Computers and Communication (ISCC)*, 2015, pp. 180–187.

[115] B. Barak, “Constant-round Coin-tossing with a Man in The Middle or Realizing The Shared Random String Model,” in *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings*, 2002, pp. 345–355.

[116] V. Ramachandran, “Detecting Arp Spoofing: An Active Technique,” in *International Conference on Information Systems Security*, 2005, pp. 239–250.

[117] S. Son, “The Hitchhiker’s Guide to DNS Cache Ppoisoning,” in *International Conference on Security and Privacy in Communication Systems*, 2010, pp. 466–483.

[118] P. De Ryck, “Secsess: Keeping Your Session Tucked Away in Your Browser,” in *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, 2015, pp. 2171–2176.

[119] K. Sonar, “Ddos Attack on Internet of Things,” *Int. J. Eng. Res. Dev.*, vol. 10, no. 13, pp. 58–63, 2014.

[120] A. Bijalwan, “Forensics of Random-udp Flooding Attacks,” *J. Networks*, vol. 10, no. 5, p. 287, 2015.

[121] M. Beaumont-Gay, “A Comparison of Syn Flood Detection Algorithms,” in *Second International Conference on Internet Monitoring and Protection (ICIMP 2007)*, 2007, p. 9.

[122] J. Erickson, *Hacking: The Art of Exploitation*. No starch press, 2008.

[123] Y. G. Dantas, “A Selective Defense for Application Layer Ddos Attacks,” in *2014 IEEE Joint Intelligence and Security Informatics Conference*, 2014, pp. 75–8.

[124] J. Krupp, “Identifying The Scan and Attack Iinfrastructures Behind Amplification Ddos Attacks,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 1426–1437.

[125] Z. Trifa, “Sybil Nodes As A Mitigation Sstrategy Against Sybil Attack,” in *Procedia Computer Science*, 2014, pp. 1135–1140.

[126] I. Mavridis, “Performance Evaluation of Cloud-Based Log File Analysis with Apache Hadoop and Apache Spark,” *ournal Syst. Softw.*, vol. 125, pp. 133–151, 2017.

[127] O. Brun, “Deep learning with dense random neural network for detecting attacks against IoT-connected home environments,” in *Procedia Computer Science*, 2018, pp. 458–463.

[128] A. Gupta, “A Big Data Analysis Framework using Apache Spark and Deep Learning,” in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, 2017, pp. 9–16.

[129] A. Khumoyun, “Spark Based Distributed Ddeep Learning Framework for Big Data Applications,” in *2016 International Conference on Information Science and Communications Technologies (ICISCT)*, 2016, pp. 1–5.

[130] S. Kak, “New Algorithms for Training Feedforward Neural Networks,” *Pattern Recognit. Lett.*, vol. 15, no. 3, pp. 295–298, 1994.

[131] C.-J. Hsieh, “Detection Ddos Attacks Based On Neural-Network Using Apache Spark,” in *2016 International Conference on Applied System Innovation (ICASI)*, 2016, pp. 1–4.

[132] A. Dawoud, “Deep Learning and Software-

Defined Networks: Towards Secure IoT Architecture," in *Internet of Things 3*, 2018, pp. 82–89.

[133] R. Kozik, "A Scalable Distributed Machine Learning Approach for Attack Detection in Edge Computing Environments," *J. Parallel Distrib. Comput.*, vol. 119, pp. 18–26, 2018.

[134] A. K. Sood, "Crimeware-as-a-service—a Survey of Commoditized Crimeware in The Underground Market," *Int. J. Crit. Infrastruct. Prot.*, vol. 6, no. 1, pp. 28–38, 2013.

[135] D. Ravi, "A Deep Learning Approach to On-node Sensor Data Analytics for Mobile or Wearable Devices," *IEEE J. Biomed. Heal. informatics*, vol. 21, no. 1, pp. 56–64, 2016.

[136] T. A. Tang, "Deep Learning Approach for Network Intrusion Ddetection in Software Defined Networking," in *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 2016, pp. 258–263.

[137] W. Wang, "Hast-ids: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," in *IEEE Access*, 2017, pp. 1792–1806.

[138] I. Kotenko, "Attack Detection in IoT Critical Infrastructures: A machine Learning and Big Data Processing Approach," in *2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, 2019.

[139] D. C. Mocanu, "Big IoT Data Mining for Real-Time Energy Disaggregation in Buildings," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016.

[140] I. Kotenko, "Framework for Mobile Internet of Things Security Monitoring Based on Big Data Processing and Machine Learning," in *IEEE Access*, 2018, pp. 72714–72723.

[141] Z. Lu, "An IoT Big Data-Oriented MapReduce Performance Prediction Eextended Model in Multiple Edge Clouds," *J. Parallel Distrib. Comput.*, pp. 316–327, 2018.