# REFERENSI TUGAS AKHIR

|  |  |  |
| --- | --- | --- |
| JUDUL PAPER | MASALAH YANG DIANGKAT | REVIEW HASIL |
| REAL-TIME INTRUSION DETECTION SYSTEM USING MULTI-AGENT SYSTEM | Tuntutan untuk mengurangi *processing time* menjadi kewajiban karena pesatnya pertumbuhan *network attack* membuat *processing* *time* IDS semakin panjang | Solusi yang diberikan adalah dengan menawarkan sistem Multi-agent (MAS-IDS). Sistem Multi-agent banyak digunakan guna meningkatkan akurasi dari sebuah IDS, namun MAS juga dapat digunakan untuk mengurangi *processing time* yang dibutuhkan.  Jumlah *agent* dapat ditambah sebanyak apapun sesuai dengan ukuran traffic data dan kapabilitas *logical processor* yang ada pada sistem, tanpa membebani kinerja host.  Hasilnya dapat mengurangi hingga 81% *processing time* pada analisa dataset KDDCUP99 dibandingkan dengan IDS tradisional |
| SCALABLE DISTRIBUTED REAL-TIME CLUSTERING FOR BIG DATA STREAMS | Volume data terus bertambah, penghasil terbesarnya adalah internat. Data yang relevan dapat di*extract*, dan dilakukan analisa, semua berkat teknologi machine learning dan data mining. Untuk mengatasi masalah volume, kecepatan dan *variety of data*, dilakukan studi tentang streaming model, dimana analisa data haruslah menggunakan *low memory* dan *item* diproses hanya sekali maka dari itu hanya informasi statistikal yang disimpan untuk mengurangi kebutuhan *storage*.  Saat ini, data tersebut di*transport* melalui advance grid dan teknologi cloud computing yang canggih, namun terdapat komponen yang kurang ditengah-tengah teknologi tersebut, yaitu teknologi distributed computing.  Stream Processing Engine, diketahui lebih fleksibel dan *powerful* jika berurusan dengan BigData. Salah satu kelebihannya adalah kemampuan untuk memproses *unbound, and infinite data flow* kapanpun dibutuhkan. | Project yang dilakukan, adalah dengan menciptakan framework untuk mengembangkan, menguji, dan menerapkan algoritma pada data stream dengan volume yang besar. Menggabungkan konsep machine learning, streaming model dan distributed computing.  Projectnya bernama SAMOA (Scalable Advanced Massive Online Analysis), dengan fokusan hasil utama adalah pada aspek *scalability* dan *clustering quality*. Hasilnya menunjukan bahwa SAMOA dapat berjalan dengan baik pada *distributed environment* dan memberikan kapabilitas scaling-up untuk mengatasi *loads* yang lebih besar |
| REAL-TIME HYBRID INTRUSION DETECTION SYSTEM USING APACHE STORM | Seiring berkembangnya internet, jumlah data semakin bertambah hingga mencapai ukuran yang sangat besar, data tersebut harus di proses oleh network IDS untuk dapat mendeteksi adanya intrusi.  Dengan ini *intrusion detection* harus dilakukan secara realtime sebelum *intruder* dapat menyebabkan kerugian, masalahnya adalah IDS sebelumnya tidak sesuai dengan kebutuhan akan *big data* | Solusi yang ditawarkan adalah dengan membangun real-time hybrid IDS menggunakan Apache Storm. Apache Storm bertindak sebagai prosesor real time *big data stream,* yang terdistribusi dan *fault tolerant*.  Hybrid detection system maksudnya adalah penggunaan dua algoritma neural network, yaitu CC4 *neural network* yang bertindak sebagai *anomaly-based detection* dan Multi Layer Perceptron neural network yang bertindak sebagai *misuse-based detection* untuk attack yang sudah diketahui.  Hasilnya adalah, output label (“attack” atau “normal”) yang memiliki akurasi 89% dengan 4.32% tingkat false positive. |
| PERFORMANCE ANALYSIS OF NETWORK INTRUSION DETECTION SCHEMES USING APACHE SPARK | Data semakin besar dan kompleks, sehinggal real time intrusion detection system harusnya dapat menghandle network traffic dengan ukuran yang besar, secepat mungkin untuk mencegah terjadinya intrusi | Lima algoritma machine learning diimplementasiikan pada sebuah IDS, yaitu Logistic regression, Support vector machines, Random forest, Gradient Boosted decision dan Naive bayes untuk mendeteksi attack. Apache spark digunakan untuk melakukan pemrosesan big data untuk mendeteksi dan menganalisa intrusi yang terjadi di trafik  Hasilnya berupa komparasi antara kelima algoritma tersebut. Aspek yang dinilai adalah training time, predicition time, akurasi, sensitivitas, dan specificity dengan menggunakan real-time dataset KDD99.  Yang memiliki performa dengan aspek di atas adalah algoritma Random Forest. |
| HADOOP BASED REAL-TIME INTRUSION DETECTION FOR HIGH-SPEED NETWORKS | Jumlah data tumbuh pesat seiring meningkatnya pengguna internet dan kecepatannya. Hal ini meningkatkan kemungkinan terjadinya intrusi yang dapat menyebabkan kerusakan finansial. Dan juga, mendeteksi intrusi pada jaringan dengan data berkecepatan tinggi (high-speed) data network merupakan hal yang sulit dilakukan. | Solusi yang ditawarkan adalah berupa IDS yang dapat bekerja di environment big data dan pada network dengan data berkecepatan tinggi.  Desain system memiiliki empat layer, yaitu *capturing layer, filtration and load balancing layer, processing layer, dan decision-making layer*. Pengujian dilakukan menggunakan DARPA dataset.  Hasilnya, dari sekiab banyak pendekatan machine learning, system yang diajukan yang paling baik performanya adalah REPTree dan J48. Dengan hasilnya dibandingkan dengan sistem intrusi yang telah ada adalah efisisensi dan akurasi yang lebih baik dengan mencapai 99.9% true positive dengan kurang dari 0.001% false positive menggunakan REPTree |
| DISTRIBUTED NETWORK TRAFFIC FEATURE EXTRACTION FOR A REAL-TIME IDS | Internet traffic meningkat dengan diikuti network attack yang juga meningkat. | Solusi yang ditawarkan adalah berupa *architecture based IDS.* Yang dapat mendeteksi anomali di jaringan secara real-time. System IDS ini memanfaatkan framework apache spark dan Netmap.  Hasilnya adalah berupa implementasi modul feature extraction ke distributed IDS dan juga hasil dari proses komputasional traffic TCP-based. Dataset yang diunakan adalah CAIDA attack dataset, dan ekstraksi fitur hanya dapat dilakukan pada network traffic untuk small organization.  Future worksnya adalah dengan memperluas scale dari sistem agar dapat mengatasi traffic untuk core dan backbone network secara efisien. Hal tersebut dapat dicapai dengan memperhatikan hal berikut ini, menambahkan jumlah node pada spark cluster dan meningkatkan proses implementasinya.  Pada implementasi sekarang, seluruh Spark SQL based queries dibuat menggunakan *single spark job*. Padahal ada banyak query yang dapat dieksekusi secara paralel. Job tersebut dapat dibagi-bagi menjadi beberapa spark jobs untuk mencapati kecepatan yang lebih tinggi. |